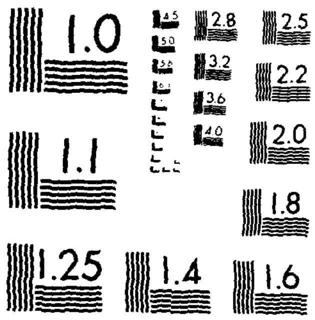
AD-A195 339 RESEARCH ON A TWO-STAGE FREE ELECTRON LASER OSCILLATOR 1/1
(U) BERKELEY RESEARCH ASSOCIATES INC CA MAR 88
BRA-88-W034R N00014-85-C-2069

UNCLASSIFIED F/G 9/3 NL

END
DATE
FILMED
8 88



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

BERKELEY RESEARCH
ASSOCIATES, INC.

BRA-88-W034R
March 1988

RESEARCH ON A TWO-STAGE FREE ELECTRON LASER OSCILLATOR

Final Report

Contract N00014-85-C-2069

for

Plasma Physics Division
Naval Research Laboratory
Washington, DC 20375

by

Berkeley Research Associates, Inc.
P.O. Box 241
Berkeley, California 94701

BRA-88-W034R
March 1988

RESEARCH ON A TWO-STAGE

FREE ELECTRON LASER OSCILLATOR

Final Report

Contract N00014-85-C-2069

for

Plasma Physics Division
Naval Research Laboratory
Washington, DC 20375

by

Berkeley Research Associates, Inc.
P.O. Box 241
Berkeley, California 94701

I. INTRODUCTION

This final report covers research carried out in conjunction with the Plasma Physics Division of the Naval Research Laboratory for the performance period 10 December 1984 to 9 December 1985 for Contract N00014-86-C-2069. The research involved the theoretical and numerical analysis of the physics of free electron lasers using relativistic particle beams. The major emphasis of the research was to obtain design criteria for the development of a two-stage FEL oscillator operating in the trapped particle mode. The present work was centered on the development of a fully relativistic, nonlinear analysis of the spatial and temporal evolution of multiple modes within a free electron laser oscillator and of a large amplitude, nonlinearly saturated state characteristic of trapped particle mode operation.

The equations solved are the Maxwell equations of electrodynamics coupled with the collisionless Boltzmann equation that describes collisionless particles under influence of the electromagnetic fields. The electromagnetic fields include the radiation fields from the FEL and the self-electric fields from the longitudinal potential due to the space charge, i.e., the dominant component of the interparticle Coulomb forces. The particle dynamics transverse to the magnetic axis are included, but gradients in the radiation fields are ignored. The electron beam equilibrium is assumed to be spatially uniform and temporally stationary. Justification and probable impact of further approximations are discussed in the technical section of this report. The approximations employed are consistent with the purpose of obtaining experimentally implementable design criteria for the FEL oscillator.

Both analytical and numerical analyses were performed. The numerical simulations are in qualitative and quantitative agreement with the analytical theories. For the FEL's of interest, the theory exhibits the same scaling as is obtained for an FEL amplifier operating in the low gain regime and the ultrahigh gain regime. The intermediate case, the moderate gain regime, is directly applicable to the experimental parameters of the FEL at NRL. For example, for a beam energy of 500 keV, a current of 100 A, radius of 0.64 cm and a wiggler length of 4.0 cm with wiggler field strength of 615 G, the theoretical expression for the threshold reflection coefficient is 0.64. The experimentally measured value is 0.65, a very

satisfactory agreement. This and other examples may be found in the NRL Memorandum Report 5679 contained in Appendix B. This NRL memorandum report has also been published in Nuclear Instruments and Methods in Physics Research **A250**, *159-167* (*1986*). Appendix A contains the FORTRAN listing of the computer codes that were written for use in the research.

II. TECHNICAL DISCUSSION

We have conducted an analytical and numerical analysis of the field evolution in a high gain free electron laser (FEL) that is operating in the oscillator configuration. This high gain oscillator provides the pump field for the second stage of a two stage free electron laser that is operating in the trapped particle regime.

The analysis contains the following attributes. The electron beam is assumed to be spatially uniform and temporally stationary. The magnetostatic wiggler field is helically symmetric and generates circularly polarized radiation with the same polarization sense as that of the electron beam motion in the magnetic field. The motion of the electrons in the combined radiation and magnetic fields produces ponderomotive bunching which is inhibited by the self consistently obtained space charge fields which arise due to the particle bunching.

In order to obtain a detailed understanding of the distinct physical processes which contribute to the operation of the FEL in this configuration, several specific computer codes were developed and are summarized in the following discussion.

The first of these is entitled MULTI.FOR and enables one to monitor the simultaneous evolution of many longitudinal modes of the radiation field. These fields are given by

$$\mathbf{A}_R(z,t) = \sum_n a_n(t) sin(k_n z) exp(i\omega_n t)\hat{e}_- + c.c. \tag{1}$$

and

$$\phi(z,t) = \sum_n \phi_{1n} sin[(k_n + k_w)z - \omega_n t] + \phi_{2n} cos[(k_n + k_w)z - \omega_n t] \tag{2}$$

where $\mathbf{A}_R(z,t)$ is the vector potential for the circularly polarized radiation, $\phi(z,t)$ is the self-consistent longitudinal space charge potential, c.c. is the complex conjugate of the first term on the right of Equation (1) and k_w is the wiggler wave number. Computationally, the decomposition of the fields can contain at most fifty modes. The operation of this code yields information on the growth properties and saturated values for the radiation field.

For the sake of completeness, computer codes were developed to evaluate the theoretical expressions for the growth rate for comparison to the simulation results. Two computer codes were developed for this purpose and they are entitled PARTEMP.FOR and MAXWTEM.FOR. In addition to being capable of evaluating the growth rate for the cold beam case, these codes have the capacity to evaluate the growth rate when the electron beam possesses a spread in parallel energy. In the code PARTEMP.FOR, the parallel temperature spread is modeled with a Lorentzian distribution in axial momentum and the associated dispersion relation is evaluated by quadratic interpolation methods.

The code MAXWTEM.FOR models the axial spread with a Maxwellion distribution. The associated dispersion relation is expressed in terms of the Fied-Conte plasma dispersion function and the growth rates are again evaluated by quadratic interpolation methods. By defining a characteristic axial spread, Δp_z, which encompasses 70% of the beam electrons, one notes that irrespective of the detailed structure of the axial distribution one obtains the same growth rate. Now, comparing this growth rate to a simulation case, as shown in Figure 1. One notes that the horizontal line representing the theoretical growth rate for the mode under consideration is in excellent agreement with the simulation results for the linear gain regime for which the comparison is appropriate.

For significant reductions in the electron beam current or wiggler field strength the growth rate for the generated radiation is also reduced. These conditions require calculations with increased accuracy, and for this reason a double precision version of the field evolution code was developed. This code is entitled SPRADB.FOR and returns all the capabilities of the single precision code MULTI.FOR.

The increased storage requirements for the double precision variables in addition to the increase cpu time for operations on these variables made long runs or large numbers of particles prohibative for code operation on the VAX 780 or 785. For these reasons a Cray FORTRAN code was developed and entitled GRAYWIG4.FOR. This code made use of the Cray processing advantages while retaining the attributes and capabilities of the code developed on the VAX. Further optimization of the code data handling capabilities were incorporated in the Cray codes entitled CMPLXAMP.FOR and RESTART.FOR.

The code COMPLXAMP.FOR made use of the complex amplitude representation of the radiation fields to evolve the system of equations. This method proved to be faster and more stable than its Cray predecessor. The final code, RESTART.FOR, was developed to take previous results from CMPLXAMP.FOR as input in order to evolve the system for longer times.

The application of these computational tools in the analysis of experimental results is presented in the attached publication in Appendix B. The quantitative and qualitative comparisons are excellent.

III. FIGURES

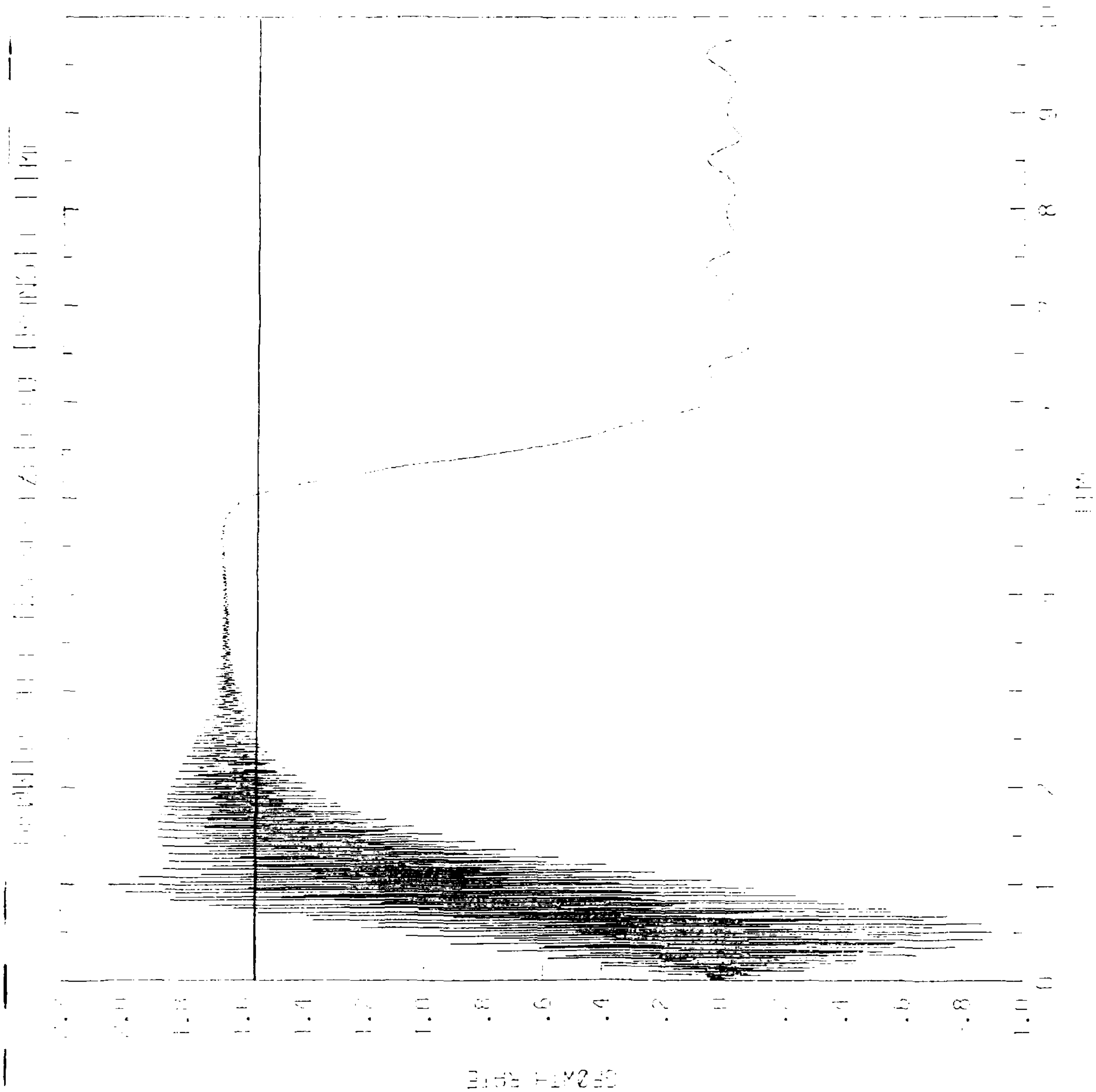


FIGURE 1

IV. APPENDICES

A. CODE LISTINGS

```
0000000000     44444444444444444444444444444444444444444444444444     0000000000
0000000000     Digital Equipment Corporation - VAX/VMS Version V4.6     0000000000
0000000000     44444444444444444444444444444444444444444444444444     0000000000
```

```
M   M   AAA   RRRR    AAA   BBBB   L      EEEEE
MM MM  A   A  R   R  A   A  B   B  L      E
M M M  A   A  R   R  A   A  B   B  L      E
M   M  A   A  RRRR   A   A  BBBB   L      EEEE
M   M  AAAAA  R R    AAAAA  B   B  L      E
M   M  A   A  R  R   A   A  B   B  L      E
M   M  A   A  R   R  A   A  BBBB   LLLLL  EEEEE

 CCCC  M   M  PPPP   L      X   X   AAA   M   M  PPPP
C      MM MM  P   P  L      X   X  A   A  MM MM  P   P
C      M M M  P   P  L       X X   A   A  M M M  P   P
C      M   M  PPPP   L        X    A   A  M   M  PPPP
C      M   M  P      L       X X   AAAAA  M   M  P
C      M   M  P      L      X   X  A   A  M   M  P
 CCCC  M   M  P      LLLLL  X   X  A   A  M   M  P

          CCCC  FFFFF  TTTTT   ;;    1
         C      F        T     ;;   11
         C      F        T           1
         C      FFFF     T     ;;    1
         C      F        T     ;;    1
      .. C      F        T      ;    1
      ..  CCCC  F        T     ;   111
```

File _VC$DRB1:[MARABLE]CMPLXAMP.CFT;1 (4034,5,0), last revised on 21-MAR-1988 14:05, is a 34 block sequential file owned by UIC [MARABLE]. The records are variable length with implied (CR) carriage control. The longest record is 72 bytes.

Job CMPLXAMP (1995) queued to LN03_QUE on 21-MAR-1988 14:08 by user MARABLE, UIC [MARABLE], under account 4790 at priority 100, started on printer LTA8: on 21-MAR-1988 14:08 from queue VC_LN03B.


```
0000000000     44444444444444444444444444444444444444444444444444     0000000000
0000000000     Digital Equipment Corporation - VAX/VMS Version V4.6     0000000000
0000000000     44444444444444444444444444444444444444444444444444     0000000000
```

```
C***   CODE TO EVALUATE TEMPORAL EVOLUTION OF THE SPRECTRA
C***   OF UNSTABLE MODES IN A HELICAL WIGGLER FREE ELECTRON LASER
C***   DELETION OF FIRST TRANSIT TIME
C***   FIELD AND PARTICLE EQUATIONS ARE EVOLVED BY ADAMS-BASHFORTH
C***   METHOD WITH INITALIZATION BY RUNGE-KUTTA METHOD
C***   REFORMULATION OF THE PARTICLE PHASE  3/13
C***      CONVERSION TO CRAY FORTRAN
C***   INCLUSION OF FREQUENCY SHIFT ERROR CHECK IN ADAMS-BASHFORTH
C***   EQUATION SOLVER
C***   REPLACE EXPRESSION FOR THE DERIVATIVES WITH THE FUNCTIONAL
C***   EVALUATION OF THE DIFFERENTIAL EQUATION
C***   COMPLEX AMPLITUDE FORMULATION OF FIELD EVOLUTION
       REAL BETAZ0,BETA0,KPOD(20),OMEGA(20),BETAZ,GAM
       REAL PSI(20,3000,5),U0(3000,5)
       REAL TEMP(3000),TEMP1(3000)
       REAL TIME(5000),PLOTA(5000,20),GROWTH(5000,20)
       REAL FREQ(5000,20),EWAV(5000),GROE(5000)
       REAL PHI1(20,5),PHI2(20,5),PHI1T(20,4),PHI2T(20,4),KWIGL
     1 ,PLAI(5000,20),AMAG(20)
       REAL KWIGR,NU,NUR,NUI,FILL,PLEE(5000,3),PLAR(5000,20)
       REAL PSIT(20,3000,4),U0T(3000,4)
       REAL K31(3000),K32(3000),K33(3000),K34(3000)
       REAL K41(3000),K42(3000),K43(3000),K44(3000)
       COMPLEX ATEMP(20),A(20,5),APP(20,5),CORA(20),AT(20,4)
       COMPLEX APT(20,4),K11(20),K12(20),K13(20),K14(20)
       COMPLEX F11(20),F12(20),F13(20),F14(20),F15(20)
       COMPLEX ALPHA1,CAPP(20)
       INTEGER F,MM,JP,J,J1,K,MAXIT
       COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
       COMMON/BLK2/PSIT,U0T,PHI1T,PHI2T,AT,APT
       COMMON/BLK3/TIME,NPART,N,RISE,TAU,BETA1,BETA2,F,ALPHA1
       COMMON/BLK4/ ALPHA2,ALPHA3,NMODE
       COMMON/BLK5/PSI,U0,PHI1,PHI2,A,APP
       PARAMETER (PI=3.1415926535)
       PS00(J1,J) = -OMEGA(J1)*FLOAT(J-1)*TAU
       TRISE(N) = 1. -RISE*(EXP((-TIME(N)+1.)/RISE) -
     1 EXP(-TIME(N)/RISE) )
       OPEN(UNIT=1,FILE='FT01',FORM='UNFORMATTED',STATUS='NEW')
       OPEN(UNIT=2,FILE='FT02',FORM='UNFORMATTED',STATUS='NEW')
1      FORMAT( ' THE PRED.-CORR. METHD FAILED TO CONVERGE ON STEP',2X,
     1   I4,' AFTER ',I4,2X,' INTERATIONS')
       WRITE(6,10)
10     FORMAT( 'INPUT NO. OF PART.,NO. OF ITERAT.,GAM,BETAWIG,KWIGR
     1   ,BUDKER,NWIG,EPS,PHASE,RISE,NPLUS,NMODE,NSEP,REF,F,FILL
     1   ,MAXIT,ERROR,ERROR2')
       READ(5,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,PHASE
     1   ,RISE,NPLUS,NMODE,NSEP,REF,F,FILL,MAXIT,ERROR,ERROR2
20     FORMAT(' INPUT DATA:NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,
     1 EPS,PHASE,RISE,NPLUS,NMODE,NSEP,REF,F,FILL,MAXIT,ERROR,ERROR2')
       WRITE(6,20)
       WRITE(6,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,PHASE,
     1   RISE,NPLUS,NMODE,NSEP,REF,F,FILL,MAXIT,ERROR,ERROR2
       KWIGL = 2.*FLOAT(NWIG)*PI
       BETA0 = SQRT(1.0 - 1.0/(GAM0*GAM0))
       BETAZ0 = SQRT(BETA0*BETA0 - BETAW*BETAW)
       NOPT = NINT(2.*FLOAT(NWIG)*BETAZ0/(1.-BETAZ0)) +NPLUS
       OMEGA(1) = (FLOAT(NOPT)*PI)/BETAZ0
       KPOD(1) = KWIGL + BETAZ0*OMEGA(1)
       DO 50 J=2,(NMODE-1)/2 + 1
       OMEGA(J) = FLOAT(NOPT+(J-1)*NSEP)*PI/BETAZ0
       KPOD(J) = KWIGL +BETAZ0*OMEGA(J)
       OMEGA(NMODE+2-J) =FLOAT(NOPT-(J-1)*NSEP)*PI/BETAZ0
       KPOD(NMODE+2-J) = KWIGL + BETAZ0*OMEGA(NMODE+2-J)
50     CONTINUE
       BETA1 = 2.*FILL*NU*KWIGL**2*BETA0*BETAW/(KWIGR**2*BETAZ0**3)
       NUR = (1.-REF)/BETAZ0
```

```
      NUI = -4.*FILL*NU*KWIGL**2*(1.-BETAW**2/2.)/(GAM0*BETAZ0*
     1  KWIGR**2*BETAZ0*OMEGA(1))
      BETA2 = 8.*NU*BETA0*KWIGL**2/(BETAZ0*KWIGR**2)
      BETA2 = 0.
      ALPHA1 = -(NUR + CMPLX(0.,1.)*NUI)
      ALPHA2 = KPOD(1)/(BETAZ0*BETAZ0)
      ALPHA3 = .25*KPOD(1)*BETAW*GAM0/BETAZ0**2
C***  INITIALIZE PHASE AND AMPLITUDE
      TIME(1) = 1.
      GROE(1) = 0.
      TAU = 1./FLOAT(NPART -1)
      TAUT = FLOAT(F)*TAU
C***  INITIALIZE EACH MODE UNDER CONSIDERATION AND FIND
C***  THE WAVE ENERGY DENSITY  IN THE INITIAL SPECTURM
      EWAV0 = 0.
      DO 60 J1=1,NMODE
      FREQ(1,J1) = 0.
      GROWTH(1,J1) = 0.
      PLOTA(1,J1) = EPS
      AMAG(J1) = EPS
      A(J1,5) = -CMPLX(0.,1.)*EPS*CEXP( CMPLX(0.,1.)*PHASE )
      AT(J1,1) = A(J1,5)
      EWAV0 = EWAV0 + (KWIGR*BETAZ0*OMEGA(J1)*AMAG(J1))**2/
     1  (4.*NU*KWIGL**2*(GAM0 -1.))
60    CONTINUE
      PLEE(1,1) = EWAV0/FILL
C***  INITIALIZE PHASES FOR THE FIRST MODE
      DO 80 J=1,NPART
      U0(J,5) = KPOD(1) - OMEGA(1)
      U0T(J,1) = KPOD(1) - OMEGA(1)
      PSI(1,J,5) = PS00(1,J) + FLOAT(NPART- J)*TAU*U0(J,5)
      PSIT(1,J,1) = PS00(1,J) + FLOAT(NPART-J)*TAU*(KPOD(1)-OMEGA(1))
80    CONTINUE
      PLEE(1,2) = 1.
      PLEE(1,3) = 1. + PLEE(1,1)
C*** INITIALIZE AMPLITUDE EVOLUTION WITH THREE POINTS FROM RUNGE-KUTTA
      EWAV(1) = EWAV0
      DO 1000 N = 2,4
      N1 = N -1
      TIME(N) = FLOAT(N1)*TAUT +1.
      DO 85 NN=2,4
      DO 90 JP =NPART-F+1,NPART
      J = JP + N1*F
      U0T(JP,NN) = KPOD(1) - OMEGA(1)
90    PSIT(1,JP,NN)= PS00(1,J) +FLOAT(NPART-JP)*TAU*U0T(JP,NN)
85    CONTINUE
      DO 1100 J1=1,NMODE
      CALL EVOLVR(J1,K11,1)
      AT(J1,2) = AT(J1,1) + TAUT*K11(J1)/2.
      APT(J1,2) = K11(J1)
      APT(J1,1) = K11(J1)
      PHI1(J1,7-N) = PHI1T(J1,1)
      PHI2(J1,7-N) = PHI2T(J1,1)
      IF(N .EQ. 2) THEN
      F11(J1) = K11(J1)
      APP(J1,5) = K11(J1)
      END IF
      IF(N .EQ. 3) F12(J1) = K11(J1)
      IF(N .EQ. 4) F13(J1) = K11(J1)
1100  CONTINUE
      DO 1200 JP=1,NPART-F
      CALL F4R(JP,K41,1)
      K31(JP) = U0T(JP,1)
      PSIT(1,JP,2) = PSIT(1,JP,1) + TAUT*K31(JP)/2.
1200  U0T(JP,2) = U0T(JP,1) + TAUT*K41(JP)/2.
      DO 1300 J1=1,NMODE
```

```
      CALL EVOLVR(J1,K12,2)
      AT(J1,3) = AT(J1,1) + TAUT*K12(J1)/2.
 1300 APT(J1,3) = K12(J1)
      DO 1400 JP=1,NPART-F
      CALL F4R(JP,K42,2)
      K32(JP) = U0T(JP,1) + TAUT*K41(JP)/2.
      PSIT(1,JP,3) = PSIT(1,JP,1) + TAUT*K32(JP)/2.
 1400 U0T(JP,3) = U0T(JP,1) + TAUT*K42(JP)/2.
      DO 1500 J1=1,NMODE
      CALL EVOLVR(J1,K13,3)
      AT(J1,4) = AT(J1,1) + TAUT*K13(J1)
 1500 APT(J1,4) = K13(J1)
      DO 1600 JP=1,NPART-F
      CALL F4R(JP,K43,3)
      K33(JP) = U0T(JP,1) + TAUT*K42(JP)/2.
      PSIT(1,JP,4) = PSIT(1,JP,1) + TAUT*K33(JP)
 1600 U0T(JP,4) = U0T(JP,1) + TAUT*K43(JP)
      EWAV(N) = 0.
      DO 1900 J1=1,NMODE
      CALL EVOLVR(J1,K14,4)
      A(J1,6-N)=AT(J1,1)+TAUT*(K11(J1)+2.*K12(J1)+2.*K13(J1)+
     1 K14(J1))/6.
      AMAG(J1) = CABS( A(J1,6-N) )
      PLOTA(N,J1) = AMAG(J1)
      APP(J1,6-N) = K14(J1)
      APT(J1,1) = APP(J1,6-N)
      AT(J1,1) = A(J1,6-N)
      FREQ(N,J1) = AIMAG( APP(J1,6-N)/A(J1,6-N) )
      PLAR(N,J1) = REAL( A(J1,6-N) )
      PLAI(N,J1) = AIMAG( A(J1,6-N) )
      EWAV(N) = EWAV(N) + KWIGR**2*((BETAZ0*OMEGA(J1)*AMAG(J1))**2 +
     1   KPOD(J1)**2*(PHI1T(J1,1)**2+PHI2T(J1,1)**2)/4.)/(4.*NU*
     1   (GAM0-1.)*KWIGL**2)
 1900 GROWTH(N,J1) = REAL( APP(J1,6-N)/A(J1,6-N) )
      GROE(N) = (EWAV(N) -EWAV0)/(TAUT*EWAV0)
      EWAV0 = EWAV(N)
      PLEE(N,1) = EWAV(N)/FILL
      PLEE(N,2) = 0.
      DO 1950 JP =1,NPART-F
      CALL F4R(JP,K44,4)
      K34(JP) = U0T(JP,1) + TAUT*K43(JP)
      U0(JP,6-N) = U0T(JP,1)+TAUT*(K41(JP)+2.*K42(JP)+2.*K43(JP)
     1   +K44(JP) )/6.
      U0T(JP,1) = U0(JP,6-N)
      BETAZ = BETAZ0*(U0(JP,1) + OMEGA(1))/KPOD(1)
      GAM = 1./SQRT(1.-BETAZ*BETAZ-BETAW*BETAW)
      PLEE(N,2) = PLEE(N,2) + (GAM - 1.)
      PSI(1,JP,6-N) = PSIT(1,JP,1) + TAUT*(K31(JP) +2.*K32(JP) +
     1   2.*K33(JP) + K34(JP) )/6.
 1950 PSIT(1,JP,1) = PSI(1,JP,6-N)
      DO 1975 JP=NPART-F+1,NPART
      U0(JP,6-N) = U0T(JP,1)
      PLEE(N,2) = PLEE(N,2) + (GAM0 - 1.)
 1975 PSI(1,JP,6-N) = PSIT(1,JP,1)
      PLEE(N,2) = PLEE(N,2)*TRISE(N)/(FLOAT(NPART)*(GAM0 -1.))
      PLEE(N,3) = PLEE(N,2) + PLEE(N,1)
 1000 CONTINUE
C***  NOW EVOLVE PARTICLES AND FIELDS WITH ADAMS-BASHFORTH PREDICTOR
C***  CORRECTOR METHOD USING THE RESULTS OF THE FOUR PREVIOUS TIMES
C***  AS INITIAL CONDITIONS
      DO 7000 N=5,NTIMES
      N1 = N -1
      TIME(N) = 1. + FLOAT(N1)*TAUT
      DO 5100 J1=1,NMODE
 5100 CALL EVOLV(J1,F14,2)
      DO 5200 JP=1,NPART-4*F
```

```
      CALL F4(JP+F,FOUT,2)
      CALL F4(JP+2*F,FOUT1,3)
      CALL F4(JP+3*F,FOUT2,4)
      CALL F4(JP+4*F,FOUT3,5)
      U0(JP,1) = U0(JP+F,2) + TAUT*(55.*FOUT -59.*FOUT1 +37.*FOUT2
     1  -9.*FOUT3)/24.
      PSI(1,JP,1) = PSI(1,JP+F,2) + TAUT*(55.*U0(JP+F,2) -
     1  59.*U0(JP+2*F,3)+37.*U0(JP+3*F,4)-9.*U0(JP+4*F,5))/24.
      TEMP(JP) = 19.*FOUT -5.*FOUT1 + FOUT2
 5200 TEMP1(JP) = 19.*U0(JP+F,2)-5.*U0(JP+2*F,3)+U0(JP+3*F,4)
      SUM  = 0.
      DO 5300 JP=NPART-4*F+1,NPART-F
      CALL F4(JP+F,FOUT,2)
      U0(JP,1) = U0(JP+F,2) + TAUT*FOUT
      BETAZ = BETAZ0*(U0(J1,1) + OMEGA(1) )/KPOD(1)
      GAM = 1./SQRT(1. -BETAZ*BETAZ -BETAW*BETAW)
      SUM = SUM + GAM -1.
 5300 PSI(1,JP,1) = PSI(1,JP+F,2)+TAUT*(U0(JP+F,2) +U0(JP,1))/2.
      DO 5400 JP = NPART-F+1,NPART
      J = JP + N1*F
      U0(JP,1) = KPOD(1) - OMEGA(1)
      SUM = SUM + GAM0 -1.
 5400 PSI(1,JP,1) = PS00(1,J)+ FLOAT(NPART-JP)*TAU*U0(JP,1)
      DO 5500 J1=1,NMODE
      A(J1,1) = A(J1,2)+TAUT*(55.*F14(J1)-59.*F13(J1)+37.*F12(J1)
     1  - 9.*F11(J1))/24.
      APP(J1,1) = F14(J1)
      ATEMP(J1) = 19.*F14(J1) - 5.*F13(J1) + F12(J1)
 5500 CONTINUE
      DO 5800 M=1,MAXIT
      DO 5700 J1=1,NMODE
      CALL EVOLV(J1,F15,1)
      CAPP(J1) = APP(J1,1)
      CORA(J1) = A(J1,1)
      A(J1,1) = A(J1,2) + TAUT*(9.*F15(J1) + ATEMP(J1))/24.
 5700 APP(J1,1) = F15(J1)
      EWAV(N) = 0.
      PLEE(N,2) = 0.
      DO 5600 JP =1,NPART-4*F
      CALL F4(JP,FOUT,1)
      U0(JP,1) = U0(JP+F,2) +TAUT*(9.*FOUT +TEMP(JP))/24.
      BETAZ = BETAZ0*(U0(J1,1) + OMEGA(1) )/KPOD(1)
      GAM = 1./SQRT(1. -BETAZ*BETAZ - BETAW*BETAW)
      PLEE(N,2) = PLEE(N,2) + GAM -1.
 5600 PSI(1,JP,1) = PSI(1,JP+F,2) +TAUT*(9.*U0(JP,1)+TEMP1(JP))/24.
      PLEE(N,2) = (PLEE(N,2) + SUM)*TRISE(N)/(FLOAT(NPART)*(GAM0-1.))
      DO 5750 J1=1,NMODE
      TEST=ABS(AIMAG(APP(J1,1)/A(J1,1))-AIMAG(CAPP(J1)/CORA(J1)))
      IF( CABS( A(J1,1)-CORA(J1))/CABS(CORA(J1)) .GT. ERROR .OR.
     1 CABS(A(J1,1)-CORA(J1))/CABS(CORA(J1)-A(J1,2)) .GT. ERROR .OR.
     1  TEST/ABS(AIMAG(CAPP(J1)/CORA(J1))) .GT. ERROR2) THEN
      GO TO 5799
      ELSE
      AMAG(J1) = CABS(A(J1,1))
      PLOTA(N,J1) = AMAG(J1)
      GROWTH(N,J1) = REAL( APP(J1,1)/A(J1,1) )
      PLAR(N,J1) = REAL( A(J1,1) )
      PLAI(N,J1) = AIMAG( A(J1,1) )
      FREQ(N,J1) = AIMAG( APP(J1,1)/A(J1,1) )
      EWAV(N) = EWAV(N) + KWIGR**2*((BETAZ0*OMEGA(J1)*AMAG(J1))**2 +
     1   KPOD(J1)**2*(PHI1(J1,1)**2+PHI2(J1,1)**2)/4.)/(4.*NU*
     1   (GAM0-1.)*KWIGL**2 )
      END IF
 5750 CONTINUE
      GROE(N) = ( EWAV(N) -EWAV0)/(TAUT*EWAV0)
      EWAV0 = EWAV(N)
```

```
      PLEE(N,1) = EWAV(N)/FILL
      PLEE(N,3) = PLEE(N,2) + PLEE(N,1)
      GO TO 5850
5799  IF(M .EQ. MAXIT)WRITE(6,1)N,M
5800  CONTINUE
 850  DO 6200 K=4,1,-1
      DO 5900 J1=1,NMODE
      A(J1,K+1) = A(J1,K)
 900  APP(J1,K+1) = APP(J1,K)
      DO 6100 JP=1,NPART
      U0(JP,K+1) = U0(JP,K)
      PSI(1,JP,K+1) = PSI(1,JP,K)
 100  CONTINUE
 200  CONTINUE
      DO 6300 J1=1,NMODE
      F11(J1) = F12(J1)
      F12(J1) = F13(J1)
6300  F13(J1) = F14(J1)
7000  CONTINUE
      WRITE(1) TIME,PLOTA,GROWTH,FREQ,EWAV,GROE,PLAR,PLAI,
     1 KPOD,OMEGA,KWIGR,NU,GAM0,BETAW,RISE,
     1 FILL,REF,EPS,PHASE,ERROR,BETAZ0,ERROR2
      WRITE(2)NWIG,NPART,F,NMODE,NPLUS,MAXIT,NSEP,NTIMES
      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
      END
      SUBROUTINE F4R(JP,K4,MM)
      REAL BETAZ0,BETAZ,GAM,KPOD(20),OMEGA(20)
      REAL PHI1T(20,4),PHI2T(20,4),PSIT(20,3000,4)
      REAL U0T(3000,4),K4(3000)
      COMPLEX AT(20,4),APT(20,4),SUM1,SUM2,NUM
      INTEGER JP,MM,NMODE
      COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
      COMMON/BLK2/PSIT,U0T,PHI1T,PHI2T,AT,APT
      COMMON/BLK4/ALPHA2,ALPHA3,NMODE
      NUM = CMPLX(0.,1.)
      BETAZ = BETAZ0*(U0T(JP,MM) + OMEGA(1) )/KPOD(1)
      GAM = 1./SQRT(1. -BETAZ*BETAZ - BETAW*BETAW)
      SUM1 = KPOD(1)*(PHI2T(1,MM)*COS(PSIT(1,JP,MM))-
     1   PHI1T(1,MM)*SIN(PSIT(1,JP,MM)))
      SUM2 = (KPOD(1)-BETAZ0*BETAZ*OMEGA(1))*(CONJG(AT(1,MM))*
     1  CEXP(NUM*PSIT(1,JP,MM))+AT(1,MM)*CEXP(-NUM*PSIT(1,JP,MM)))
     1  -NUM*BETAZ0*BETAZ*KPOD(1)*(CONJG(APT(1,MM))*CEXP(
     1   NUM*PSIT(1,JP,MM))-APT(1,MM)*CEXP(-NUM*PSIT(1,JP,MM)))
      DO 100 J1=2,NMODE
      SUM1 = SUM1 + KPOD(J1)*(PHI2T(J1,MM)*COS(PSIT(J1,JP,MM))
     1   -PHI1T(J1,MM)*SIN(PSIT(J1,JP,MM)))
      SUM2 = SUM2 +(KPOD(J1)-BETAZ0*BETAZ*OMEGA(J1))*(
     1  CONJG(AT(J1,MM))*CEXP(NUM*PSIT(J1,JP,MM))+AT(J1,MM)*
     1  CEXP(-NUM*PSIT(J1,JP,MM)))-NUM*BETAZ0*BETAZ*KPOD(J1)*
     1  (CONJG(APT(J1,MM))*CEXP(NUM*PSIT(J1,JP,MM))-APT(J1,MM)*
     1  CEXP(-NUM*PSIT(J1,JP,MM)))
100   CONTINUE
      K4(JP) = ALPHA2*REAL(SUM1)*(1.-BETAZ*BETAZ)/GAM +
     1   ALPHA3*REAL(SUM2)/(GAM*GAM)

      RETURN
      END
      SUBROUTINE F4(JP,FOUT,MM)
      REAL BETAZ,BETAZ0,GAM,KPOD(20),OMEGA(20)
      REAL PSI(20,3000,5),U0(3000,5),PHI1(20,5),PHI2(20,5),FOUT
      COMPLEX A(20,5),APP(20,5),SUM1,SUM2,NUM
      INTEGER JP,MM,NMODE
      COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
      COMMON/BLK5/PSI,U0,PHI1,PHI2,A,APP
      COMMON/BLK4/ALPHA2,ALPHA3,NMODE
```

```
      NUM = CMPLX(0.,1.)
      BETAZ = BETAZ0*(U0(JP,MM) + OMEGA(1) )/KPOD(1)
      GAM = 1./SQRT(1.-BETAZ*BETAZ - BETAW*BETAW)
      SUM1 =KPOD(1)*(PHI2(1,MM)*COS(PSI(1,JP,MM))
     1   -PHI1(1,MM)*SIN(PSI(1,JP,MM)))
      SUM2 =(KPOD(1)-BETAZ0*BETAZ*OMEGA(1))*(CONJG(A(1,MM))*
     1  CEXP(NUM*PSI(1,JP,MM))+A(1,MM)*CEXP(-NUM*PSI(1,JP,MM)))
     1  -NUM*BETAZ0*BETAZ*KPOD(1)*(CONJG(APP(1,MM))*CEXP(NUM*
     1  PSI(1,JP,MM))-APP(1,MM)*CEXP(-NUM*PSI(1,JP,MM)))
      DO 100 J1=2,NMODE
      SUM1=SUM1 +KPOD(J1)*(PHI2(J1,MM)*COS(PSI(J1,JP,MM))
     1   -PHI1(J1,MM)*SIN(PSI(J1,JP,MM)))
      SUM2=SUM2+(KPOD(J1)-BETAZ0*BETAZ*OMEGA(J1))*(CONJG(A(J1,MM))
     1 *CEXP(NUM*PSI(J1,JP,MM))+A(J1,MM)*CEXP(-NUM*PSI(J1,JP,MM)))
     1 -NUM*BETAZ0*BETAZ*KPOD(J1)*(CONJG(APP(J1,MM))*CEXP(NUM*
     1 PSI(J1,JP,MM))-APP(J1,MM)*CEXP(-NUM*PSI(J1,JP,MM)))
100   CONTINUE
      FOUT = ALPHA2*REAL(SUM1)*(1.-BETAZ*BETAZ)/GAM +
     1   ALPHA3*REAL(SUM2)/(GAM*GAM)
      RETURN
      END
      SUBROUTINE EVOLVR(J1,K1,MM)
      REAL BETAZ0,BETAZ,GAM,KPOD(20),OMEGA(20)
      REAL PHI1T(20,4),PHI2T(20,4),PSIT(20,3000,4),U0T(3000,4)
      REAL TIME(5000)
      COMPLEX AT(20,4),APT(20,4),K1(20),DUM1,NUM,ALPHA1
      INTEGER J1,MM,N,F,NPART
      COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
      COMMON/BLK2/PSIT,U0T,PHI1T,PHI2T,AT,APT
      COMMON/BLK3/TIME,NPART,N,RISE,TAU,BETA1,BETA2,F,ALPHA1
      NUM = CMPLX(0.,1.)
      DUM1 = CMPLX(0.,0.)
      DUM2 = 0.
      DUM3 = 0.
      DO 100 JP =1,NPART
      J = JP + (N-1)*F
      TRISE = 1. - EXP(-FLOAT(J-1)*TAU/RISE)
      BETAZ = BETAZ0*(U0T(JP,MM) + OMEGA(1) )/KPOD(1)
      GAM = 1./SQRT(1.-BETAZ*BETAZ - BETAW*BETAW)
      PSIT(J1,JP,MM)=KPOD(J1)*(PSIT(1,JP,MM)+OMEGA(1)*TIME(N))/KPOD(1)
     1   - OMEGA(J1)*TIME(N)
      DUM1= DUM1 + CEXP(NUM*PSIT(J1,JP,MM))*TAU*TRISE*GAM0/GAM
      DUM2 = DUM2 + COS(PSIT(J1,JP,MM))*TAU*TRISE
100   DUM3 = DUM3 + SIN(PSIT(J1,JP,MM))*TAU*TRISE
      K1(J1) = ALPHA1*AT(J1,MM) + BETA1*DUM1/OMEGA(J1)
      PHI1T(J1,MM) = - BETA2*DUM2/KPOD(J1)**2
      PHI2T(J1,MM) = - BETA2*DUM3/KPOD(J1)**2
      RETURN
      END
      SUBROUTINE EVOLV(J1,K1,MM)
      REAL BETAZ0,BETAZ,GAM,KPOD(20),OMEGA(20)
      REAL PHI1(20,5),PHI2(20,5),PSI(20,3000,5),U0(3000,5)
      REAL TIME(5000)
      COMPLEX A(20,5),APP(20,5),K1(20),DUM1,ALPHA1,NUM
      INTEGER J1,MM,N,F,NPART
      COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
      COMMON/BLK5/PSI,U0,PHI1,PHI2,A,APP
      COMMON/BLK3/TIME,NPART,N,RISE,TAU,BETA1,BETA2,F,ALPHA1
      NUM = CMPLX(0.,1.)
      DUM1 = CMPLX(0.,0.)
      DUM2 = 0.
      DUM3 = 0.
      DO 100 JP =1,NPART
      J = JP + (N-1)*F
      TRISE = 1. - EXP(-FLOAT(J-1)*TAU/RISE)
      BETAZ = BETAZ0*(U0(JP,MM) + OMEGA(1) )/KPOD(1)
```

```
      GAM = 1./SQRT(1.-BETAZ*BETAZ - BETAW*BETAW)
      PSI(J1,JP,MM)=KPOD(J1)*(PSI(1,JP,MM)+OMEGA(1)*TIME(N))/KPOD(1)
     1   - OMEGA(J1)*TIME(N)
      DUM1 = DUM1 + CEXP(NUM*PSI(J1,JP,MM))*TAU*TRISE*GAM0/GAM
      DUM2 = DUM2 + COS(PSI(J1,JP,MM))*TAU*TRISE
00    DUM3 = DUM3 + SIN(PSI(J1,JP,MM))*TAU*TRISE
      K1(J1) = ALPHA1*A(J1,MM) + BETA1*DUM1/OMEGA(J1)
      PHI1(J1,MM) = - BETA2*DUM2/KPOD(J1)**2
      PHI2(J1,MM) = - BETA2*DUM3/KPOD(J1)**2
      RETURN
      END
```

```
NNNNNNNNNN     3333333333333333333333333333333333333333333333333333     NNNNNNNNNN
NNNNNNNNNN     Digital Equipment Corporation - VAX/VMS Version V4.6     NNNNNNNNNN
NNNNNNNNNN     3333333333333333333333333333333333333333333333333333     NNNNNNNNNN


                 M   M   AAA   RRRR    AAA   BBBB   L      EEEEE
                 MM MM  A   A  R   R  A   A  B   B  L      E
                 M M M  A   A  R   R  A   A  B   B  L      E
                 M   M  A   A  RRRR   A   A  BBBB   L      EEEE
                 M   M  AAAAA  R R    AAAAA  B   B  L      E
                 M   M  A   A  R  R   A   A  B   B  L      E
                 M   M  A   A  R   R  A   A  BBBB   LLLLL  EEEEE


                  CCCC  RRRR    AAA   Y   Y  W   W   III    GGGG
                 C      R   R  A   A  Y   Y  W   W    I    G
                 C      R   R  A   A   Y Y   W   W    I    G
                 C      RRRR   A   A    Y    W   W    I    G
                 C      R R    AAAAA    Y    W W W    I    G GGG
                 C      R  R   A   A    Y    WW WW    I    G   G
                  CCCC  R   R  A   A    Y    W   W   III    GGG

                            CCCC  FFFFF  TTTTT    ;;     1
                           C      F        T      ;;    11
                           C      F        T             1
                           C      FFFF     T      ;;     1
                           C      F        T      ;;     1
                        .. C      F        T       ;     1
                        ..  CCCC  F        T      ;     111
```

ile _VC$DRB1:[MARABLE]CRAYWIG.CFT;1 (4036,4,0), last revised on 21-MAR-1988 14:
5, is a 37 block sequential file owned by UIC [MARABLE]. The records are varia
ble length with implied (CR) carriage control. The longest record is 72 bytes.

ob CRAYWIG (1997) queued to LN03_QUE on 21-MAR-1988 14:09 by user MARABLE, UIC
[MARABLE], under account 4790 at priority 100, started on printer LTA7: on
21-MAR-1988 14:09 from queue VC_LN03A.


```
NNNNNNNNNN     3333333333333333333333333333333333333333333333333333     NNNNNNNNNN
NNNNNNNNNN     Digital Equipment Corporation - VAX/VMS Version V4.6     NNNNNNNNNN
NNNNNNNNNN     3333333333333333333333333333333333333333333333333333     NNNNNNNNNN
```

```
C***    CODE TO EVALUATE TEMPORAL EVOLUTION OF THE SPRECTRA
C***    OF UNSTABLE MC.ES IN A HELICAL WIGGLER FREE ELECTRON LASER
C***    DELETION OF FIRST TRANSIT TIME
C***    FIELD AND PARTICLE EQUATIONS ARE EVOLVED BY ADAMS-BASHFORTH
C***    METHOD WITH INITALIZATION BY RUNGE-KUTTA METHOD
C***    REFORMULATION OF THE PARTICLE PHASE  3/13
C***       CONVERSION TO CRAY FORTRAN
        REAL BETAZ0,BETA0,KPOD(20),OMEGA(20),BETAZ,GAM
        REAL CTHET(20),PSI(20,3000,5),U0(3000,5)
        REAL TEMP(3000),TEMP1(3000),ATEMP(20),ATEMP1(20),THETA(20,5)
        REAL TP(20,5),TIME(5000),PLOTA(5000,20),GROWTH(5000,20)
        REAL FREQ(5000,20),EWAV(5000),GROE(5000),A(20,5),APP(20,5)
        REAL PHI1(20,5),PHI2(20,5),PHI1T(20,4),PHI2T(20,4),KWIGL
     1  ,CORA(20),PLAI(5000,20)
        REAL KWIGR,NU,NUR,NUI,FILL,PLEE(5000,3),PLAR(5000,20)
        REAL PSIT(20,3000,4),U0T(3000,4),THETAT(20,4),TPT(20,4),AT(20,4)
        REAL APT(20,4),K11(20),K12(20),K13(20),K14(20),K21(20),K22(20)
        REAL K23(20),K24(20),K31(3000),K32(3000),K33(3000)
        REAL K34(3000),K41(3000),K42(3000),K43(3000),K44(3000)
        REAL F11(20),F12(20),F13(20),F14(20),F15(20),F21(20),F22(20)
        REAL F23(20),F24(20),F25(20)
        INTEGER F,MM,JP,J,J1,K,MAXIT
        COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
        COMMON/BLK2/PSIT,U0T,PHI1T,PHI2T,AT,THETAT,APT,TPT
        COMMON/BLK3/TIME,NPART,N,RISE,TAU,BETA1,BETA2,F,NUI,NUR
        COMMON/BLK4/ ALPHA1,ALPHA2,NMODE
        COMMON/BLK5/PSI,U0,PHI1,PHI2,A,THETA,APP,TP
        PARAMETER (PI=3.1415926535)
        PS00(J1,J) = -OMEGA(J1)*FLOAT(J-1)*TAU
        TRISE(N) = 1. -RISE*(EXP((-TIME(N)+1.)/RISE) -1.)
        OPEN(UNIT=1,FILE='FT01',STATUS='NEW')
        OPEN(UNIT=2,FILE='FT02',STATUS='NEW')
1       FORMAT( ' THE PRED.-CORR. METHD FAILED TO CONVERGE ON STEP',2X,
     1    I4,' AFTER ',I4,2X,' INTERATIONS')
        WRITE(6,10)
10      FORMAT( 'INPUT NO. OF PART.,NO. OF ITERAT.,GAM,BETAWIG,KWIGR
     1    ,BUDKER,NWIG,EPS,PHASE,RISE,NPLUS,NMODE,NSEP,REF,F,FILL
     1    ,MAXIT,ERROR,ERROR2')
        READ(5,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,PHASE
     1    ,RISE,NPLUS,NMODE,NSEP,REF,F,FILL,MAXIT,ERROR,ERROR2
20      FORMAT(' INPUT DATA:NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,
     1  EPS,PHASE,RISE,NPLUS,NMODE,NSEP,REF,F,FILL,MAXIT,ERROR,ERROR2')
        WRITE(6,20)
        WRITE(6,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,PHASE,
     1    RISE,NPLUS,NMODE,NSEP,REF,F,FILL,MAXIT,ERROR,ERROR2
        KWIGL = 2.*FLOAT(NWIG)*PI
        BETA0 = SQRT(1.0 - 1.0/(GAM0*GAM0))
        BETAZ0 = SQRT(BETA0*BETA0 - BETAW*BETAW)
        NOPT = NINT(2.*FLOAT(NWIG)*BETAZ0/(1.-BETAZ0)) +NPLUS
        OMEGA(1) = (FLOAT(NOPT)*PI)/BETAZ0
        KPOD(1) = KWIGL + BETAZ0*OMEGA(1)
        DO 50 J=2,(NMODE-1)/2 + 1
        OMEGA(J) = FLOAT(NOPT+(J-1)*NSEP)*PI/BETAZ0
        KPOD(J) = KWIGL +BETAZ0*OMEGA(J)
        OMEGA(NMODE+2-J) =FLOAT(NOPT-(J-1)*NSEP)*PI/BETAZ0
        KPOD(NMODE+2-J) = KWIGL + BETAZ0*OMEGA(NMODE+2-J)
50      CONTINUE
        BETA1 = 2.*FILL*NU*KWIGL**2*BETA0*BETAW/(KWIGR**2*BETAZ0**3)
        NUR = (1.-REF)/BETAZ0
        NUI = -4.*FILL*NU*KWIGL**2*(1.-BETAW**2/2.)/(GAM0*BETAZ0*
     1    KWIGR**2*BETAZ0*OMEGA(1))
        BETA2 = 8.*NU*BETA0*KWIGL**2/(BETAZ0*KWIGR**2)
        BETA2 = 0.
        ALPHA1 = KPOD(1)/BETAZ0
        ALPHA2 = .5*BETAW*GAM0*KPOD(1)/BETAZ0**2
C***    INITIALIZE PHASE AND AMPLITUDE
```

```
      TIME(1) = 1.
      GROE(1) = 0.
      TAU = 1./FLOAT(NPART -1)
      TAUT = FLOAT(F)*TAU
***   INITIALIZE EACH MODE UNDER CONSIDERATION AND FIND
***   THE WAVE ENERGY DENSITY  IN THE INITIAL SPECTURM
      EWAV0 = 0.
      DO 60 J1=1,NMODE
      PLOTA(1,J1) = EPS
      A(J1,5) = EPS
      AT(J1,1) = EPS
      THETA(J1,5) = PHASE
      THETAT(J1,1) = PHASE
      EWAV0 = EWAV0 + (KWIGR*BETAZ0*OMEGA(J1)*A(J1,5))**2/
     1  (4.*NU*KWIGL**2*(GAM0 -1.))
0     CONTINUE
      PLEE(1,1) = EWAV0/FILL
C***  INITIALIZE PHASES FOR THE FIRST MODE
      DO 80 J=1,NPART
      U0(J,5) = KPOD(1) - OMEGA(1)
      U0T(J,1) = KPOD(1) - OMEGA(1)
      PSI(1,J,5) = PS00(1,J) + FLOAT(NPART- J)*TAU*U0(J,5)
      PSIT(1,J,1) = PS00(1,J) + FLOAT(NPART-J)*TAU*(KPOD(1)-OMEGA(1))
0     CONTINUE
      PLEE(1,2) = 1.
      PLEE(1,3) = 1. + PLEE(1,1)
***  INITIALIZE AMPLITUDE EVOLUTION WITH THREE POINTS FROM RUNGE-KUTTA
      EWAV(1) = EWAV0
      DO 1000 N = 2,4
      N1 = N -1
      TIME(N) = FLOAT(N1)*TAUT +1.
      DO 85 NN=2,4
      DO 90 JP =NPART-F+1,NPART
      J = JP + N1*F
      U0T(JP,NN) = KPOD(1) - OMEGA(1)
90    PSIT(1,JP,NN)= PS00(1,J) +FLOAT(NPART-JP)*TAU*U0T(JP,NN)
5     CONTINUE
      DO 1100 J1=1,NMODE
      CALL EVOLVR(J1,K21,K11,1)
      TPT(J1,2) = K21(J1)
      TPT(J1,1) = K21(J1)
      THETAT(J1,2) = THETAT(J1,1) + TAUT*K21(J1)/2.
      AT(J1,2) = AT(J1,1) + TAUT*K11(J1)/2.
      APT(J1,2) = K11(J1)
      APT(J1,1) = K11(J1)
      PHI1(J1,7-N) = PHI1T(J1,1)
      PHI2(J1,7-N) = PHI2T(J1,1)
      IF(N .EQ. 2) THEN
      F11(J1) = K11(J1)
      F21(J1) = K21(J1)
      TP(J1,5) = K21(J1)
      APP(J1,5) = K11(J1)
      FREQ(1,J1) = TPT(J1,1)
      GROWTH(1,J1) = APP(J1,5)/A(J1,5)
      END IF
      IF(N .EQ. 3) THEN
      F12(J1) = K11(J1)
      F22(J1) = K21(J1)
      END IF
      IF(N .EQ. 4) THEN
      F13(J1) = K11(J1)
      F23(J1) = K21(J1)
      END IF
1100  CONTINUE
      DO 1200 JP=1,NPART-F
      CALL F4R(JP,K41,1)
```

```
      K31(JP) = U0T(JP,1)
      PSIT(1,JP,2) = PSIT(1,JP,1) + TAUT*K31(JP)/2.
 1200 U0T(JP,2) = U0T(JP,1) + TAUT*K41(JP)/2.
      DO 1300 J1=1,NMODE
      CALL EVOLVR(J1,K22,K12,2)
      TPT(J1,3) = K22(J1)
      THETAT(J1,3) = THETAT(J1,1) + TAUT*K22(J1)/2.
      AT(J1,3) = AT(J1,1) + TAUT*K12(J1)/2.
 1300 APT(J1,3) = K12(J1)
      DO 1400 JP=1,NPART-F
      CALL F4R(JP,K42,2)
      K32(JP) = U0T(JP,1) + TAUT*K41(JP)/2.
      PSIT(1,JP,3) = PSIT(1,JP,1) + TAUT*K32(JP)/2.
 1400 U0T(JP,3) = U0T(JP,1) + TAUT*K42(JP)/2.
      DO 1500 J1=1,NMODE
      CALL EVOLVR(J1,K23,K13,3)
      TPT(J1,4) = K23(J1)
      THETAT(J1,4) = THETAT(J1,1) + TAUT*K23(J1)
      AT(J1,4) = AT(J1,1) + TAUT*K13(J1)
 1500 APT(J1,4) = K13(J1)
      DO 1600 JP=1,NPART-F
      CALL F4R(JP,K43,3)
      K33(JP) = U0T(JP,1) + TAUT*K42(JP)/2.
      PSIT(1,JP,4) = PSIT(1,JP,1) + TAUT*K33(JP)
 1600 U0T(JP,4) = U0T(JP,1) + TAUT*K43(JP)
      DO 1700 J1=1,NMODE
 1700 CALL EVOLVR(J1,K24,K14,4)
      DO 1800 JP=1,NPART-F
      CALL F4R(JP,K44,4)
 1800 K34(JP) = U0T(JP,1) + TAUT*K43(JP)
      EWAV(N) = 0.
      DO 1900 J1=1,NMODE
      A(J1,6-N)=AT(J1,1)+TAUT*(K11(J1)+2.*K12(J1)+2.*K13(J1)+
     1 K14(J1))/6.
      PLOTA(N,J1) = A(J1,6-N)
      APP(J1,6-N) = (A(J1,6-N) - AT(J1,1))/TAUT
      APT(J1,1) = APP(J1,6-N)
      AT(J1,1) = A(J1,6-N)
      THETA(J1,6-N)=THETAT(J1,1)+TAUT*(K21(J1)+2.*K22(J1)+2.*K23(J1)
     1   + K24(J1))/6.
      TP(J1,6-N) = (THETA(J1,6-N) - THETAT(J1,1))/TAUT
      TPT(J1,1) = TP(J1,6-N)
      THETAT(J1,1) = THETA(J1,6-N)
      FREQ(N,J1) = TP(J1,6-N)
      PLAR(N,J1) = AT(J1,1)*SIN(THETAT(J1,1))
      PLAI(N,J1) = -AT(J1,1)*COS(THETAT(J1,1))
      EWAV(N) = EWAV(N) + KWIGR**2*((BETAZ0*OMEGA(J1)*A(J1,6-N))**2 +
     1   KPOD(J1)**2*(PHI1T(J1,1)**2+PHI2T(J1,1)**2)/4.)/(4.*NU*
     1   (GAM0-1.)*KWIGL**2)
 1900 GROWTH(N,J1) = APP(J1,6-N)/A(J1,6-N)
      GROE(N) = (EWAV(N) -EWAV0)/(TAUT*EWAV0)
      EWAV0 = EWAV(N)
      PLEE(N,1) = EWAV(N)/FILL
      PLEE(N,2) = 0.
      DO 1950 JP =1,NPART-F
      U0(JP,6-N) = U0T(JP,1)+TAUT*(K41(JP)+2.*K42(JP)+2.*K43(JP)
     1   +K44(JP) )/6.
      U0T(JP,1) = U0(JP,6-N)
      BETAZ = BETAZ0*(U0(JP,1) + OMEGA(1))/KPOD(1)
      GAM = 1./SQRT(1.-BETAZ*BETAZ-BETAW*BETAW)
      PLEE(N,2) = PLEE(N,2) + (GAM - 1.)
      PSI(1,JP,6-N) = PSIT(1,JP,1) + TAUT*(K31(JP) +2.*K32(JP) +
     1   2.*K33(JP) + K34(JP) )/6.
 1950 PSIT(1,JP,1) = PSI(1,JP,6-N)
      DO 1975 JP=NPART-F+1,NPART
      U0(JP,6-N) = U0T(JP,1)
```

```
      PLEE(N,2) = PLEE(N,2) + (GAM0 - 1.)
1975  PSI(1,JP,6-N) = PSIT(1,JP,1)
      PLEE(N,2) = PLEE(N,2)*TRISE(N)/(FLOAT(NPART)*(GAM0 -1.))
      PLEE(N,3) = PLEE(N,2) + PLEE(N,1)
1000  CONTINUE
C***  NOW EVOLVE PARTICLES AND FIELDS WITH ADAMS-BASHFORTH PREDICTOR
C***  CORRECTOR METHOD USING THE RESULTS OF THE FOUR PREVIOUS TIMES
C***  AS INITIAL CONDITIONS
      DO 7000 N=5,NTIMES
      N1 = N -1
      TIME(N) = 1. + FLOAT(N1)*TAUT
      DO 5100 J1=1,NMODE
5100  CALL EVOLV(J1,F24,F14,2)
      DO 5200 JP=1,NPART-4*F
      CALL F4(JP+F,FOUT,2)
      CALL F4(JP+2*F,FOUT1,3)
      CALL F4(JP+3*F,FOUT2,4)
      CALL F4(JP+4*F,FOUT3,5)
      U0(JP,1) = U0(JP+F,2) + TAUT*(55.*FOUT -59.*FOUT1 +37.*FOUT2
     1  -9.*FOUT3)/24.
      PSI(1,JP,1) = PSI(1,JP+F,2) + TAUT*(55.*U0(JP+F,2) -
     1  59.*U0(JP+2*F,3)+37.*U0(JP+3*F,4)-9.*U0(JP+4*F,5))/24.
      TEMP(JP) = 19.*FOUT -5.*FOUT1 + FOUT2
5200  TEMP1(JP) = 19.*U0(JP+F,2)-5.*U0(JP+2*F,3)+U0(JP+3*F,4)
      SUM  = 0.
      DO 5300 JP=NPART-4*F+1,NPART-F
      CALL F4(JP+F,FOUT,2)
      U0(JP,1) = U0(JP+F,2) + TAUT*FOUT
      BETAZ = BETAZ0*(U0(J1,1) + OMEGA(1) )/KPOD(1)
      GAM = 1./SQRT(1. -BETAZ*BETAZ -BETAW*BETAW)
      SUM = SUM + GAM -1.
5300  PSI(1,JP,1) = PSI(1,JP+F,2)+TAUT*(U0(JP+F,2) +U0(JP,1))/2.
      DO 5400 JP = NPART-F+1,NPART
      J = JP + N1*F
      U0(JP,1) = KPOD(1) - OMEGA(1)
      SUM = SUM + GAM0 -1.
5400  PSI(1,JP,1) = PS00(1,J)+ FLOAT(NPART-JP)*TAU*U0(JP,1)
      DO 5500 J1=1,NMODE
      A(J1,1) = A(J1,2)+TAUT*(55.*F14(J1)-59.*F13(J1)+37.*F12(J1)
     1  - 9.*F11(J1))/24.
      THETA(J1,1) = THETA(J1,2) + TAUT*(55.*F24(J1)-59.*F23(J1)
     1  +37.*F22(J1) - 9.*F21(J1))/24.
      APP(J1,1) = ( A(J1,1) -A(J1,2) )/TAUT
      TP(J1,1) = ( THETA(J1,1) - THETA(J1,2) )/TAUT
      ATEMP(J1) = 19.*F14(J1) - 5.*F13(J1) + F12(J1)
      ATEMP1(J1) = 19.*F24(J1) - 5.*F23(J1) + F22(J1)
5500  CONTINUE
      DO 5800 M=1,MAXIT
      DO 5700 J1=1,NMODE
      CALL EVOLV(J1,F25,F15,1)
      CORA(J1) = A(J1,1)
      AT(J1,1) = A(J1,2) + TAUT*(9.*F15(J1) + ATEMP(J1))/24.
       CTHET(J1) = THETA(J1,1)
      THETAT(J1,1) = THETA(J1,2) + TAUT*(9.*F25(J1) + ATEMP1(J1))/24.
      TPT(J1,1) = (THETAT(J1,1) -THETA(J1,2) )/TAUT
5700  APT(J1,1) = ( AT(J1,1) - A(J1,2))/TAUT
      EWAV(N) = 0.
      PLEE(N,2) = 0.
      DO 5600 JP =1,NPART-4*F
      CALL F4(JP,FOUT,1)
      U0(JP,1) = U0(JP+F,2) +TAUT*(9.*FOUT +TEMP(JP))/24.
      BETAZ = BETAZ0*(U0(J1,1) + OMEGA(1) )/KPOD(1)
      GAM = 1./SQRT(1. -BETAZ*BETAZ - BETAW*BETAW)
      PLEE(N,2) = PLEE(N,2) + GAM -1.
5600  PSI(1,JP,1) = PSI(1,JP+F,2) +TAUT*(9.*U0(JP,1)+TEMP1(JP))/24.
      PLEE(N,2) = (PLEE(N,2) + SUM)*TRISE(N)/(FLOAT(NPART)*(GAM0-1.))
```

```
      DO 5750 J1=1,NMODE
      IF( ABS( AT(J1,1)-CORA(J1))/ABS(CORA(J1)) .GT. ERROR .OR.
     1 ABS(THETAT(J1,1)-CTHET(J1))/ABS(CTHET(J1)) .GT. ERROR2)THEN
      DO 5725 J2=1,NMODE
      A(J2,1) = AT(J2,1)
      THETA(J2,1) = THETAT(J2,1)
      APP(J2,1) = APT(J2,1)
5725  TP(J2,1) = TPT(J2,1)
      GO TO 5799
      ELSE
      PLOTA(N,J1) = AT(J1,1)
      GROWTH(N,J1) = 2.*( AT(J1,1)-A(J1,2) )/(TAUT*(AT(J1,1)+A(J1,2)))
      PLAR(N,J1) = AT(J1,1)*SIN(THETAT(J1,1))
      PLAI(N,J1) = -AT(J1,1)*COS(THETAT(J1,1))
      FREQ(N,J1) = TP(J1,1)
      EWAV(N) = EWAV(N) + KWIGR**2*((BETAZ0*OMEGA(J1)*AT(J1,1))**2 +
     1   KPOD(J1)**2*(PHI1(J1,1)**2+PHI2(J1,1)**2)/4.)/(4.*NU*
     1   (GAM0-1.)*KWIGL**2 )
      END IF
5750  CONTINUE
      GROE(N) = ( EWAV(N) -EWAV0)/(TAUT*EWAV0)
      EWAV0 = EWAV(N)
      PLEE(N,1) = EWAV(N)/FILL
      PLEE(N,3) = PLEE(N,2) + PLEE(N,1)
      GO TO 5850
5799  IF(M .EQ. MAXIT)WRITE(6,1)N,M
5800  CONTINUE
5850  DO 6200 K=4,1,-1
      DO 5900 J1=1,NMODE
      A(J1,K+1) = A(J1,K)
      APP(J1,K+1) = APP(J1,K)
      THETA(J1,K+1) = THETA(J1,K)
5900  TP(J1,K+1) = TP(J1,K)
      DO 6100 JP=1,NPART
      U0(JP,K+1) = U0(JP,K)
      PSI(1,JP,K+1) = PSI(1,JP,K)
6100  CONTINUE
6200  CONTINUE
      DO 6300 J1=1,NMODE
      F11(J1) = F12(J1)
      F12(J1) = F13(J1)
      F13(J1) = F14(J1)
      F21(J1) = F22(J1)
      F22(J1) = F23(J1)
6300  F23(J1) = F24(J1)
7000  CONTINUE
     WRITE(1,*) TIME,PLOTA,GROWTH,FREQ,EWAV,GROE,PLAR,PLAI,
     1 KPOD,OMEGA,KWIGR,NU,GAM0,BETAW,RISE,
     1 FILL,REF,EPS,PHASE,ERROR,BETAZ0,ERROR2
      WRITE(2,*)NWIG,NPART,F,NMODE,NPLUS,MAXIT,NSEP,NTIMES
      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
      END
      SUBROUTINE F4R(JP,K4,MM)
      REAL BETAZ0,BETAZ,GAM,KPOD(20),OMEGA(20)
      REAL AT(20,4),APT(20,4),PHI1T(20,4),PHI2T(20,4),PSIT(20,3000,4)
      REAL U0T(3000,4),THETAT(20,4),TPT(20,4),K4(3000)
      INTEGER JP,MM,NMODE
      COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
      COMMON/BLK2/PSIT,U0T,PHI1T,PHI2T,AT,THETAT,APT,TPT
      COMMON/BLK4/ALPHA1,ALPHA2,NMODE
      BETAZ = BETAZ0*(U0T(JP,MM) + OMEGA(1) )/KPOD(1)
      GAM = 1./SQRT(1. -BETAZ*BETAZ - BETAW*BETAW)
      SUM1 = KPOD(1)*(PHI2T(1,MM)*COS(PSIT(1,JP,MM)-THETAT(1,MM))-
     1   PHI1T(1,MM)*SIN(PSIT(1,JP,MM)-THETAT(1,MM)))
      SUM2 = (KPOD(1)-BETAZ0*BETAZ*(OMEGA(1)+TPT(1,MM)))*AT(1,MM)*
```

```
     1    SIN(PSIT(1,JP,MM)-THETAT(1,MM))-BETAZ0*BETAZ*APT(1,MM)*
     1    COS(PSIT(1,JP,MM)-THETAT(1,MM))
          DO 100 J1=2,NMODE
          SUM1 = SUM1 + KPOD(J1)*(PHI2T(J1,MM)*COS(PSIT(J1,JP,MM)-
     1    THETAT(J1,MM))-PHI1T(J1,MM)*SIN(PSIT(J1,JP,MM)-THETAT(J1,MM)))
          SUM2 = SUM2 +(KPOD(J1)-BETAZ0*BETAZ*(OMEGA(J1)+TPT(J1,MM)))*
     1    AT(J1,MM)*SIN(PSIT(J1,JP,MM)-THETAT(J1,MM))-BETAZ0*BETAZ*
     1    APT(J1,MM)*COS(PSIT(J1,JP,MM)-THETAT(J1,MM))
 00       CONTINUE
          K4(JP) = ALPHA1*SUM1*(1.-BETAZ*BETAZ)/GAM + ALPHA2*SUM2/(GAM*
     1    GAM)

          RETURN
          END
          SUBROUTINE F4(JP,FOUT,MM)
          REAL BETAZ,BETAZ0,GAM,KPOD(20),OMEGA(20)
          REAL PSI(20,3000,5),U0(3000,5),PHI1(20,5),PHI2(20,5),A(20,5),
     1    APP(20,5),TP(20,5),THETA(20,5),FOUT
          INTEGER JP,MM,NMODE
          COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
          COMMON/BLK5/PSI,U0,PHI1,PHI2,A,THETA,APP,TP
          COMMON/BLK4/ALPHA1,ALPHA2,NMODE
          BETAZ = BETAZ0*(U0(JP,MM) + OMEGA(1) )/KPOD(1)
          GAM = 1./SQRT(1.-BETAZ*BETAZ - BETAW*BETAW)
          SUM1 =KPOD(1)*(PHI2(1,MM)*COS(PSI(1,JP,MM)-THETA(1,MM))
     1    -PHI1(1,MM)*SIN(PSI(1,JP,MM)-THETA(1,MM)))
          SUM2 =(KPOD(1)-BETAZ0*BETAZ*(OMEGA(1)+TP(1,MM)))*A(1,MM)*
     1    SIN(PSI(1,JP,MM)-THETA(1,MM))-BETAZ0*BETAZ*APP(1,MM)*
     1    COS(PSI(1,JP,MM)-THETA(1,MM))
          DO 100 J1=2,NMODE
          SUM1=SUM1 +KPOD(J1)*(PHI2(J1,MM)*COS(PSI(J1,JP,MM)-THETA(J1,MM))
     1    -PHI1(J1,MM)*SIN(PSI(J1,JP,MM)-THETA(J1,MM)))
          SUM2=SUM2+(KPOD(J1)-BETAZ0*BETAZ*(OMEGA(J1)+TP(J1,MM)))*A(J1,MM)
     1    *SIN(PSI(J1,JP,MM)-THETA(J1,MM))-BETAZ0*BETAZ*APP(J1,MM)*
     1     COS(PSI(J1,JP,MM)-THETA(J1,MM))
100       CONTINUE
          FOUT = ALPHA1*SUM1*(1.-BETAZ*BETAZ)/GAM + ALPHA2*SUM2/(GAM*
     1    GAM)
          RETURN
          END
          SUBROUTINE EVOLVR(J1,K2,K1,MM)
          REAL BETAZ0,BETAZ,GAM,KPOD(20),OMEGA(20)
          REAL AT(20,4),PHI1T(20,4),PHI2T(20,4),PSIT(20,3000,4),
     1    U0T(3000,4)
          REAL K2(20),K1(20),TIME(5000),THETAT(20,4),APT(20,4),TPT(20,4)
          REAL NUI,NUR
          INTEGER J1,MM,N,F,NPART
          COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
          COMMON/BLK2/PSIT,U0T,PHI1T,PHI2T,AT,THETAT,APT,TPT
          COMMON/BLK3/TIME,NPART,N,RISE,TAU,BETA1,BETA2,F,NUI,NUR
          DUM1 = 0.
          DUM2 = 0.
          DUM3 = 0.
          DUM4 = 0.
          DO 100 JP =1,NPART
          J = JP + (N-1)*F
          TRISE = 1. - EXP(-FLOAT(J-1)*TAU/RISE)
          BETAZ = BETAZ0*(U0T(JP,MM) + OMEGA(1) )/KPOD(1)
          GAM = 1./SQRT(1.-BETAZ*BETAZ - BETAW*BETAW)
          PSIT(J1,JP,MM)=KPOD(J1)*(PSIT(1,JP,MM)+OMEGA(1)*TIME(N))/KPOD(1)
     1    - OMEGA(J1)*TIME(N)
          DUM1 = DUM1+COS(PSIT(J1,JP,MM)-THETAT(J1,MM))*TAU*TRISE*GAM0/GAM
          DUM2 = DUM2+SIN(PSIT(J1,JP,MM)-THETAT(J1,MM))*TAU*TRISE*GAM0/GAM
          DUM3 = DUM3 + COS(PSIT(J1,JP,MM)-THETAT(J1,MM))*TAU*TRISE
 00       DUM4 = DUM4 + SIN(PSIT(J1,JP,MM)-THETAT(J1,MM))*TAU*TRISE
          K1(J1) = -NUR*AT(J1,MM)/2.- BETA1*DUM2/OMEGA(J1)
```

```
      K2(J1) = -NUI/2. + BETA1*DUM1/(AT(J1,MM)*OMEGA(J1))
      PHI1T(J1,MM) = - BETA2*DUM3/KPOD(J1)**2
      PHI2T(J1,MM) = - BETA2*DUM4/KPOD(J1)**2
      RETURN
      END
      SUBROUTINE EVOLV(J1,K2,K1,MM)
      REAL BETAZ0,BETAZ,GAM,KPOD(20),OMEGA(20)
      REAL A(20,5),PHI1(20,5),PHI2(20,5),PSI(20,3000,5),U0(3000,5)
      REAL K2(20),K1(20),TIME(5000),THETA(20,5),APP(20,5),TP(20,5)
      REAL NUI,NUR
      INTEGER J1,MM,N,F,NPART
      COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
      COMMON/BLK5/PSI,U0,PHI1,PHI2,A,THETA,APP,TP
      COMMON/BLK3/TIME,NPART,N,RISE,TAU,BETA1,BETA2,F,NUI,NUR
      DUM1 = 0.
      DUM2 = 0.
      DUM3 = 0.
      DUM4 = 0.
      DO 100 JP =1,NPART
      J = JP + (N-1)*F
      TRISE = 1. - EXP(-FLOAT(J-1)*TAU/RISE)
      BETAZ = BETAZ0*(U0(JP,MM) + OMEGA(1) )/KPOD(1)
      GAM = 1./SQRT(1.-BETAZ*BETAZ - BETAW*BETAW)
      PSI(J1,JP,MM)=KPOD(J1)*(PSI(1,JP,MM)+OMEGA(1)*TIME(N))/KPOD(1)
     1  - OMEGA(J1)*TIME(N)
      DUM1 = DUM1 + COS(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE*GAM0/GAM
      DUM2 = DUM2 + SIN(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE*GAM0/GAM
      DUM3 = DUM3 + COS(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE
100   DUM4 = DUM4 + SIN(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE
      K1(J1) = -NUR*A(J1,MM)/2.- BETA1*DUM2/OMEGA(J1)
      K2(J1) = -NUI/2. + BETA1*DUM1/(A(J1,MM)*OMEGA(J1))
      PHI1(J1,MM) = - BETA2*DUM3/KPOD(J1)**2
      PHI2(J1,MM) = - BETA2*DUM4/KPOD(J1)**2
      RETURN
      END
```

```
PPPPPPPPPP    555555555555555555555555555555555555555555555555555    PPPPPPPPPP
PPPPPPPPPP    Digital Equipment Corporation - VAX/VMS Version V4.6    PPPPPPPPPP
PPPPPPPPPP    555555555555555555555555555555555555555555555555555    PPPPPPPPPP

           M   M   AAA   RRRR    AAA   BBBB   L      EEEEE
           MM MM  A   A  R   R  A   A  B   B  L      E
           M M M  A   A  R   R  A   A  B   B  L      E
           M   M  A   A  RRRR   A   A  BBBB   L      EEEE
           M   M  AAAAA  R R    AAAAA  B   B  L      E
           M   M  A   A  R  R   A   A  B   B  L      E
           M   M  A   A  R   R  A   A  BBBB   LLLLL  EEEEE

        CCCC  RRRR    AAA   Y   Y  W   W   III    GGGG  4   4
       C      R   R  A   A  Y   Y  W   W    I    G      4   4
       C      R   R  A   A   Y Y   W   W    I    G      4   4
       C      RRRR   A   A    Y    W   W    I    G      44444
       C      R R    AAAAA    Y    W W W    I    G GGG      4
       C      R  R   A   A    Y    WW WW    I    G   G      4
        CCCC  R   R  A   A    Y    W   W   III    GGG       4

                      CCCC  FFFFF  TTTTT     ;;     1
                     C      F        T       ;;    11
                     C      F        T              1
                     C      FFFF     T       ;;     1
                     C      F        T       ;;     1
                 ..  C      F        T        ;     1
                 ..   CCCC  F        T       ;     111
```

File _VC$DRB1:[MARABLE]CRAYWIG4.CFT;1 (4037,4,0), last revised on 21-MAR-1988 14:06, is a 33 block sequential file owned by UIC [MARABLE]. The records are variable length with implied (CR) carriage control. The longest record is 72 bytes.

Job CRAYWIG4 (1998) queued to LN03_QUE on 21-MAR-1988 14:09 by user MARABLE, UIC [MARABLE], under account 4790 at priority 100, started on printer LTA8: on 21-MAR-1988 14:09 from queue VC_LN03B.


```
PPPPPPPPPP    555555555555555555555555555555555555555555555555555    PPPPPPPPPP
PPPPPPPPPP    Digital Equipment Corporation - VAX/VMS Version V4.6    PPPPPPPPPP
PPPPPPPPPP    555555555555555555555555555555555555555555555555555    PPPPPPPPPP
```

```
C***    CODE TO EVALUATE TEMPORAL EVOLUTION OF THE SPRECTRA
C***    OF UNSTABLE MODES IN A HELICAL WIGGLER FREE ELECTRON LASER
C***    DELETION OF FIRST TRANSIT TIME
C***    FIELD AND PARTICLE EQUATIONS ARE EVOLVED BY ADAMS-BASHFORTH
C***    METHOD WITH INITALIZATION BY RUNGE-KUTTA METHOD
C***    REFORMULATION OF THE PARTICLE PHASE  3/13
C***        CONVERSION TO CRAY FORTRAN
C***    INCLUSION OF FREQUENCY SHIFT ERROR CHECK IN ADAMS-BASHFORTH
C***    EQUATION SOLVER
C***    REPLACE EXPRESSION FOR THE DERIVATIVES WITH THE FUNCTIONAL
C***    EVALUATION OF THE DIFFERENTIAL EQUATION
C***    PLOT DATA AFTER EVERY 10 CALCULATIONS
C***    OUTPUT DATA IF TIME LIMIT IS APPROACHED
C***    MODIFICATIONS TO PRODUCE RESTART DATA  8/1
        REAL BETAZ0,BETA0,KPOD(20),OMEGA(20),BETAZ,GAM
        REAL CTHET(20),PSI(20,3000,5),U0(3000,5),TIM
        REAL TEMP(3000),TEMP1(3000),ATEMP(20),ATEMP1(20),THETA(20,5)
        REAL TP(20,5),TIME(5000),TPLOT(6)
        REAL FREQ(5000,20),EWAV(5000),A(20,5),APP(20,5)
        REAL PHI1(20,5),PHI2(20,5),KWIGL,PSII(600,6),U0I(600,6)
     1  ,CORA(20),TPC(20),PLAI(5000,20)
        REAL KWIGR,NU,NUR,NUI,FILL,PLAR(5000,20)
        REAL K11(20),K12(20),K13(20),K14(20),K21(20),K22(20)
        REAL K23(20),K24(20),K31(3000),K32(3000),K33(3000)
        REAL K34(3000),K41(3000),K42(3000),K43(3000),K44(3000)
        REAL F11(20),F12(20),F13(20),F14(20),F15(20),F21(20),F22(20)
        REAL F23(20),F24(20),F25(20)
        INTEGER F,MM,JP,J,J1,K,MAXIT,TLIM,NCOUNT,NLAST
        COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
        COMMON/BLK3/TIM,NPART,N,RISE,TAU,BETA1,BETA2,F,NUI,NUR
        COMMON/BLK4/ ALPHA1,ALPHA2,NMODE
        COMMON/BLK5/PSI,U0,PHI1,PHI2,A,THETA,APP,TP
        PARAMETER (PI=3.1415926535)
        PS00(J1,J) = -OMEGA(J1)*FLOAT(J-1)*TAU
        TRISE(N) = 1. -RISE*(EXP((-TIM+1.)/RISE) -1.)
        OPEN(UNIT=1,FILE='FT01',FORM='UNFORMATTED',STATUS='NEW')
        OPEN(UNIT=2,FILE='FT02',FORM='UNFORMATTED',STATUS='NEW')
        OPEN(UNIT=3,FILE='FT03',FORM='UNFORMATTED',STATUS='NEW')
1       FORMAT( ' THE PRED.-CORR. METHD FAILED TO CONVERGE ON STEP',2X,
     1    I7,' AFTER ',I4,2X,' INTERATIONS',2X,'ON MODE ',I3)
        WRITE(6,10)
10      FORMAT( 'INPUT NO. OF PART.,NO. OF ITERAT.,GAM,BETAWIG,KWIGR
     1    ,BUDKER,NWIG,EPS,PHASE,RISE,NPLUS,NMODE,NSEP,REF,F,FILL
     1    ,MAXIT,ERROR,ERROR2,TLIM')
        READ(5,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,PHASE
     1    ,RISE,NPLUS,NMODE,NSEP,REF,F,FILL,MAXIT,ERROR,ERROR2,TLIM
20      FORMAT(' INPUT DATA:NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,
     1  EPS,PHASE,RISE,NPLUS,NMODE,NSEP,REF,F,FILL,MAXIT,ERROR,ERROR2')
        WRITE(6,20)
        WRITE(6,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,PHASE,
     1    RISE,NPLUS,NMODE,NSEP,REF,F,FILL,MAXIT,ERROR,ERROR2,TLIM
        NCOUNT = 1
        KWIGL = 2.*FLOAT(NWIG)*PI
        BETA0 = SQRT(1.0 - 1.0/(GAM0*GAM0))
        BETAZ0 = SQRT(BETA0*BETA0 - BETAW*BETAW)
        NOPT = NINT(2.*FLOAT(NWIG)*BETAZ0/(1.-BETAZ0)) +NPLUS
        OMEGA(1) = (FLOAT(NOPT)*PI)/BETAZ0
        KPOD(1) = KWIGL + BETAZ0*OMEGA(1)
        DO 50 J=2,(NMODE-1)/2 + 1
        OMEGA(J) = FLOAT(NOPT+(J-1)*NSEP)*PI/BETAZ0
        KPOD(J) = KWIGL +BETAZ0*OMEGA(J)
        OMEGA(NMODE+2-J) =FLOAT(NOPT-(J-1)*NSEP)*PI/BETAZ0
        KPOD(NMODE+2-J) = KWIGL + BETAZ0*OMEGA(NMODE+2-J)
50      CONTINUE
        BETA1 = 2.*FILL*NU*KWIGL**2*BETA0*BETAW/(KWIGR**2*BETAZ0**3)
        NUR = (1.-REF)/BETAZ0
```

```
      NUI = -4.*FILL*NU*KWIGL**2*(1.-BETAW**2/2.)/(GAM0*BETAZ0*
     1  KWIGR**2*BETAZ0*OMEGA(1))
      BETA2 = 8.*NU*BETA0*KWIGL**2/(BETAZ0*KWIGR**2)
      BETA2 = 0.
      ALPHA1 = KPOD(1)/BETAZ0
      ALPHA2 = .5*BETAW*GAM0*KPOD(1)/BETAZ0**2
C***  INITIALIZE PHASE AND AMPLITUDE
      TIME(1) = 1.
      TAU = 1./FLOAT(NPART -1)
      TAUT = FLOAT(F)*TAU
C***  INITIALIZE EACH MODE UNDER CONSIDERATION AND FIND
C***  THE WAVE ENERGY DENSITY  IN THE INITIAL SPECTURM
      EWAV0 = 0.
      DO 60 J1=1,NMODE
      PLAR(1,J1) = EPS*SIN(PHASE)
      PLAI(1,J1) = -EPS*COS(PHASE)
      A(J1,5) = EPS
      THETA(J1,5) = PHASE
      EWAV0 = EWAV0 + (KWIGR*BETAZ0*OMEGA(J1)*A(J1,5))**2/
     1  (4.*NU*KWIGL**2*(GAM0 -1.))
60    CONTINUE
C***  INITIALIZE PHASES FOR THE FIRST MODE
      DO 80 J=1,NPART
      U0(J,5) = KPOD(1) - OMEGA(1)
      PSI(1,J,5) = PS00(1,J) + FLOAT(NPART- J)*TAU*U0(J,5)
      IF( MOD(J,5) .EQ. 0) THEN
      JJ = INT(J/5)
      U0I(JJ,NCOUNT) = U0(J,5)
      PSII(JJ,NCOUNT) = PSI(1,J,5)
      END IF
80    CONTINUE
      TPLOT(NCOUNT) = 1.
C*** INITIALIZE AMPLITUDE EVOLUTION WITH THREE POINTS FROM RUNGE-KUTTA
      EWAV(1) = EWAV0
      DO 1000 N = 2,4
      N1 = N -1
      TIM = FLOAT(N1)*TAUT +1.
      DO 1100 J1=1,NMODE
      CALL EVOLV(J1,K21,K11,7-N)
      TP(J1,6-N) = K21(J1)
      TP(J1,7-N) = K21(J1)
      APP(J1,6-N) = K11(J1)
      APP(J1,7-N) = K11(J1)
      THETA(J1,6-N) = THETA(J1,7-N) + TAUT*K21(J1)/2.
      A(J1,6-N) = A(J1,7-N) + TAUT*K11(J1)/2.
      IF(N .EQ. 2) THEN
      F11(J1) = K11(J1)
      F21(J1) = K21(J1)
      FREQ(1,J1) = K21(J1)
      END IF
      IF(N .EQ. 3) THEN
      F12(J1) = K11(J1)
      F22(J1) = K21(J1)
      END IF
      IF(N .EQ. 4) THEN
      F13(J1) = K11(J1)
      F23(J1) = K21(J1)
      END IF
1100  CONTINUE
      DO 1200 JP=1,NPART-INT(F/2)
      CALL F4R(JP+INT(F/2),K41,7-N)
      K31(JP) = U0(JP+INT(F/2),7-N)
      PSI(1,JP,6-N) = PSI(1,JP+INT(F/2),7-N) + TAUT*K31(JP)/2.
1200  U0(JP,6-N) = U0(JP+INT(F/2),7-N) + TAUT*K41(JP)/2.
      DO 1250 JP=NPART-INT(F/2) +1,NPART
      J = JP + (N-1)*F
```

```
      U0(JP,6-N) = KPOD(1) - OMEGA(1)
1250  PSI(1,JP,6-N) = PS00(1,J) +FLOAT(NPART-JP)*TAU*U0(JP,6-N)
      DO 1300 J1=1,NMODE
      CALL EVOLV(J1,K22,K12,6-N)
      TP(J1,5-N) = K22(J1)
      APP(J1,6-N) = K12(J1)
      THETA(J1,5-N) = THETA(J1,7-N) + TAUT*K22(J1)/2.
1300  A(J1,5-N) = A(J1,7-N) + TAUT*K12(J1)/2.
      DO 1400 JP=1,NPART-INT(F/2)
      CALL F4R(JP+INT(F/2),K42,6-N)
      K32(JP) = U0(JP+INT(F/2),7-N) + TAUT*K41(JP)/2.
      PSI(1,JP,5-N) = PSI(1,JP+INT(F/2),7-N) + TAUT*K32(JP)/2.
1400  U0(JP,5-N) = U0(JP+INT(F/2),7-N) + TAUT*K42(JP)/2.
      DO 1450 JP=NPART +1-INT(F/2),NPART
      J= JP + (N-1)*F
      U0(JP,5-N) = KPOD(1) - OMEGA(1)
1450  PSI(1,JP,5-N) = PS00(1,J) + FLOAT(NPART-JP)*TAU*U0(JP,6-N)
      DO 1500 J1=1,NMODE
      CALL EVOLV(J1,K23,K13,5-N)
      TP(J1,6-N) = K23(J1)
      THETA(J1,6-N) = THETA(J1,7-N) + TAUT*K23(J1)
      A(J1,6-N) = A(J1,7-N) + TAUT*K13(J1)
1500  APP(J1,6-N) = K13(J1)
      DO 1600 JP=1,NPART-F
      CALL F4R(JP+F,K43,5-N)
      K33(JP) = U0(JP+F,7-N) + TAUT*K42(JP)/2.
      PSI(1,JP,6-N) = PSI(1,JP+F,7-N) + TAUT*K33(JP)
1600  U0(JP,6-N) = U0(JP+F,7-N) + TAUT*K43(JP)
      DO 1650 JP=NPART+1-F,NPART
      J = JP + (N-1)*F
      U0(JP,6-N) = KPOD(1) - OMEGA(1)
1650  PSI(1,JP,6-N) = PS00(1,J) + FLOAT(NPART-JP)*TAU*U0(JP,6-N)
      DO 1700 J1=1,NMODE
1700  CALL EVOLV(J1,K24,K14,6-N)
      DO 1800 JP=1,NPART-F
      CALL F4(JP+F,FOUT,6-N)
      K44(JP) = FOUT
1800  K34(JP) = U0(JP+F,7-N) + TAUT*K43(JP)
      DO 1900 J1=1,NMODE
      A(J1,6-N)=A(J1,7-N)+TAUT*(K11(J1)+2.*K12(J1)+2.*K13(J1)+
     1 K14(J1))/6.
      APP(J1,6-N) = K14(J1)
      THETA(J1,6-N)=THETA(J1,7-N)+TAUT*(K21(J1)+2.*K22(J1)+2.*K23(J1)
     1   + K24(J1))/6.
      TP(J1,6-N) = K24(J1)
1900   CONTINUE
      DO 1950 JP =1,NPART-F
      U0(JP,6-N) = U0(JP,7-N)+TAUT*(K41(JP)+2.*K42(JP)+2.*K43(JP)
     1   +K44(JP) )/6.
      PSI(1,JP,6-N) = PSI(1,JP,7-N) + TAUT*(K31(JP) +2.*K32(JP) +
     1   2.*K33(JP) + K34(JP) )/6.
1950   CONTINUE
      DO 1975 JP=NPART-F+1,NPART
      J = JP + (N-1)*F
      U0(JP,6-N) = KPOD(1) - OMEGA(1)
1975  PSI(1,JP,6-N) = PS00(1,J) + FLOAT(NPART-JP)*TAU*U0(JP,6-N)
1000  CONTINUE
C***  NOW EVOLVE PARTICLES AND FIELDS WITH ADAMS-BASHFORTH PREDICTOR
C***  CORRECTOR METHOD USING THE RESULTS OF THE FOUR PREVIOUS TIMES
C***  AS INITIAL CONDITIONS
      DO 7000 N=5,NTIMES
      N1 = N -1
      TIM = 1. + FLOAT(N1)*TAUT
      DO 5100 J1=1,NMODE
5100  CALL EVOLV(J1,F24,F14,2)
      DO 5200 JP=1,NPART-4*F
```

```
      CALL F4(JP+F,FOUT,2)
      CALL F4(JP+2*F,FOUT1,3)
      CALL F4(JP+3*F,FOUT2,4)
      CALL F4(JP+4*F,FOUT3,5)
      U0(JP,1) = U0(JP+F,2) + TAUT*(55.*FOUT -59.*FOUT1 +37.*FOUT2
     1   -9.*FOUT3)/24.
      PSI(1,JP,1) = PSI(1,JP+F,2) + TAUT*(55.*U0(JP+F,2) -
     1   59.*U0(JP+2*F,3)+37.*U0(JP+3*F,4)-9.*U0(JP+4*F,5))/24.
      TEMP(JP) = 19.*FOUT -5.*FOUT1 + FOUT2
5200  TEMP1(JP) = 19.*U0(JP+F,2)-5.*U0(JP+2*F,3)+U0(JP+3*F,4)
      DO 5300 JP=NPART-4*F+1,NPART-F
      CALL F4(JP+F,FOUT,2)
      U0(JP,1) = U0(JP+F,2) + TAUT*FOUT
5300  PSI(1,JP,1) = PSI(1,JP+F,2)+TAUT*(U0(JP+F,2) +U0(JP,1))/2.
      DO 5400 JP = NPART-F+1,NPART
      J = JP + N1*F
      U0(JP,1) = KPOD(1) - OMEGA(1)
5400  PSI(1,JP,1) = PS00(1,J)+ FLOAT(NPART-JP)*TAU*U0(JP,1)
      DO 5500 J1=1,NMODE
      A(J1,1) = A(J1,2)+TAUT*(55.*F14(J1)-59.*F13(J1)+37.*F12(J1)
     1   - 9.*F11(J1))/24.
      THETA(J1,1) = THETA(J1,2) + TAUT*(55.*F24(J1)-59.*F23(J1)
     1   +37.*F22(J1) - 9.*F21(J1))/24.
      APP(J1,1) = F14(J1)
C***  THETA PRIME IS THE AVERAGE OF THE DISCRETE AND FUNCTIONAL
C***  VALUES OF THE DERIVATIVE
      TP(J1,1) = F24(J1)
      ATEMP(J1) = 19.*F14(J1) - 5.*F13(J1) + F12(J1)
      ATEMP1(J1) = 19.*F24(J1) - 5.*F23(J1) + F22(J1)
5500  CONTINUE
      DO 5800 M=1,MAXIT
      DO 5700 J1=1,NMODE
      CALL EVOLV(J1,F25,F15,1)
      CORA(J1) = A(J1,1)
      A(J1,1) = A(J1,2) + TAUT*(9.*F15(J1) + ATEMP(J1))/24.
      CTHET(J1) = THETA(J1,1)
      THETA(J1,1) = THETA(J1,2) + TAUT*(9.*F25(J1) + ATEMP1(J1))/24.
      TPC(J1) = F25(J1)
      TP(J1,1) = F25(J1)
5700  APP(J1,1) = F15(J1)
      IF( MOD(N,10) .EQ. 0 )THEN
      NPLOT = INT(N/10) + 1
      TIME(NPLOT) = TIM
      EWAV(NPLOT) = 0.
       END IF
      DO 5600 JP =1,NPART-4*F
      CALL F4(JP,FOUT,1)
      U0(JP,1) = U0(JP+F,2) +TAUT*(9.*FOUT +TEMP(JP))/24.
5600  PSI(1,JP,1) = PSI(1,JP+F,2) +TAUT*(9.*U0(JP,1)+TEMP1(JP))/24.
      DO 5750 J1=1,NMODE
      IF( ABS( A(J1,1)-CORA(J1))/ABS(CORA(J1)) .GT. ERROR .OR.
     1  ABS(THETA(J1,1)-CTHET(J1))/ABS(CTHET(J1)) .GT. ERROR .OR.
     1  ABS(TP(J1,1)-TPC(J1))/ABS(TP(J1,1)) .GT. ERROR2)THEN
      GO TO 5799
      ELSE
      IF( MOD(N,10) .EQ. 0)THEN
      NPLOT = INT(N/10) + 1
      PLAR(NPLOT,J1) = A(J1,1)*SIN(THETA(J1,1))
      PLAI(NPLOT,J1) = -A(J1,1)*COS(THETA(J1,1))
      FREQ(NPLOT,J1) = TP(J1,1)
      EWAV(NPLOT) = EWAV(NPLOT) + KWIGR**2*((BETAZ0*OMEGA(J1)*
     1   A(J1,1))**2 + KPOD(J1)**2*(PHI1(J1,1)**2+PHI2(J1,1)**2)
     1   /4.)/(4.*NU*(GAM0-1.)*KWIGL**2 )
      END IF
      END IF
5750  CONTINUE
```

```
      GO TO 5850
799   IF(M .EQ. MAXIT)WRITE(6,1)N,M
5800  CONTINUE
5850  DO 6200 K=4,1,-1
      DO 5900 J1=1,NMODE
      A(J1,K+1) = A(J1,K)
      APP(J1,K+1) = APP(J1,K)
      THETA(J1,K+1) = THETA(J1,K)
5900  TP(J1,K+1) = TP(J1,K)
      DO 6100 JP=1,NPART-F
      U0(JP+F,K+1) = U0(JP,K)
      PSI(1,JP+F,K+1) = PSI(1,JP,K)
6100  CONTINUE
6200  CONTINUE
      DO 6300 J1=1,NMODE
      F11(J1) = F12(J1)
      F12(J1) = F13(J1)
      F13(J1) = F14(J1)
      F21(J1) = F22(J1)
      F22(J1) = F23(J1)
6300  F23(J1) = F24(J1)
      CALL SECOND(CPU)
      IF( CPU .GE. .97*FLOAT(TLIM) ) GO TO 7001
7000  CONTINUE
7001  NLAST = N
      JJ=0
      TPLOT(NCOUNT+1) = TIM
      DO 7002 JP=5,NPART,5
      JJ = JJ + 1
      U0I(JJ,NCOUNT+1) = U0(JP,2)
7002  PSII(JJ,NCOUNT+1) = PSI(1,JP,2)
      WRITE(1) TIME,FREQ,EWAV,PLAR,PLAI,KPOD,OMEGA,KWIGR,
     1 NU,GAM0,BETAW,RISE,FILL,REF,EPS,PHASE,
     1 ERROR,BETAZ0,ERROR2,PSII,U0I,TPLOT,TLIM,BETA0
      WRITE(2)NWIG,NPART,F,NMODE,NPLUS,MAXIT,NSEP,NTIMES
     1 ,NCOUNT,NLAST
      WRITE(3) PSI,U0,F11,F12,F13,F21,F22,F23,A,APP,THETA,
     1 TP,PHI1,PHI2
      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
      CLOSE(UNIT=3)
      END
      SUBROUTINE F4R(JP,K4,MM)
      REAL BETAZ0,BETAZ,GAM,KPOD(20),OMEGA(20)
      REAL A(20,5),PHI1(20,5),PHI2(20,5),PSI(20,3000,5)
      REAL U0(3000,5),THETA(20,5),TP(20,5),K4(3000),APP(20,5)
      INTEGER JP,MM,NMODE
      COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
      COMMON/BLK5/PSI,U0,PHI1,PHI2,A,THETA,APP,TP
      COMMON/BLK4/ALPHA1,ALPHA2,NMODE
      BETAZ = BETAZ0*(U0(JP,MM) + OMEGA(1) )/KPOD(1)
      GAM = SQRT((1.+(GAM0*BETAW)**2)/(1.-BETAZ*BETAZ) )
      SUM1 = KPOD(1)*(PHI2(1,MM)*COS(PSI(1,JP,MM)-THETA(1,MM))-
     1   PHI1(1,MM)*SIN(PSI(1,JP,MM)-THETA(1,MM)))
      SUM2 = (KPOD(1)-BETAZ0*BETAZ*(OMEGA(1)+TP(1,MM)))*A(1,MM)*
     1   SIN(PSI(1,JP,MM)-THETA(1,MM))-BETAZ0*BETAZ*APP(1,MM)*
     1   COS(PSI(1,JP,MM)-THETA(1,MM))
      DO 100 J1=2,NMODE
      SUM1 = SUM1 + KPOD(J1)*(PHI2(J1,MM)*COS(PSI(J1,JP,MM)-
     1   THETA(J1,MM))-PHI1(J1,MM)*SIN(PSI(J1,JP,MM)-THETA(J1,MM)))
      SUM2 = SUM2 +(KPOD(J1)-BETAZ0*BETAZ*(OMEGA(J1)+TP(J1,MM)))*
     1   A(J1,MM)*SIN(PSI(J1,JP,MM)-THETA(J1,MM))-BETAZ0*BETAZ*
     1   APP(J1,MM)*COS(PSI(J1,JP,MM)-THETA(J1,MM))
100   CONTINUE
      K4(JP-INT(F/2)) = ALPHA1*SUM1*(1.-BETAZ*BETAZ)/GAM + ALPHA2
     1   *SUM2/(GAM*GAM)
```

```
      RETURN
      END
      SUBROUTINE F4(JP,FOUT,MM)
      REAL BETAZ,BETAZ0,GAM,KPOD(20),OMEGA(20)
      REAL PSI(20,3000,5),U0(3000,5),PHI1(20,5),PHI2(20,5),A(20,5),
     1  APP(20,5),TP(20,5),THETA(20,5),FOUT
      INTEGER JP,MM,NMODE
      COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
      COMMON/BLK5/PSI,U0,PHI1,PHI2,A,THETA,APP,TP
      COMMON/BLK4/ALPHA1,ALPHA2,NMODE
      BETAZ = BETAZ0*(U0(JP,MM) + OMEGA(1) )/KPOD(1)
      GAM = SQRT( (1.+(GAM0*BETAW)**2)/(1.-BETAZ*BETAZ))
      SUM1 =KPOD(1)*(PHI2(1,MM)*COS(PSI(1,JP,MM)-THETA(1,MM))
     1  -PHI1(1,MM)*SIN(PSI(1,JP,MM)-THETA(1,MM)))
      SUM2 =(KPOD(1)-BETAZ0*BETAZ*(OMEGA(1)+TP(1,MM)))*A(1,MM)*
     1  SIN(PSI(1,JP,MM)-THETA(1,MM))-BETAZ0*BETAZ*APP(1,MM)*
     1  COS(PSI(1,JP,MM)-THETA(1,MM))
      DO 100 J1=2,NMODE
      SUM1=SUM1 +KPOD(J1)*(PHI2(J1,MM)*COS(PSI(J1,JP,MM)-THETA(J1,MM))
     1  -PHI1(J1,MM)*SIN(PSI(J1,JP,MM)-THETA(J1,MM)))
      SUM2=SUM2+(KPOD(J1)-BETAZ0*BETAZ*(OMEGA(J1)+TP(J1,MM)))*A(J1,MM)
     1  *SIN(PSI(J1,JP,MM)-THETA(J1,MM))-BETAZ0*BETAZ*APP(J1,MM)*
     1   COS(PSI(J1,JP,MM)-THETA(J1,MM))
100   CONTINUE
      FOUT = ALPHA1*SUM1*(1.-BETAZ*BETAZ)/GAM + ALPHA2*SUM2/(GAM*
     1  GAM)
      RETURN
      END
      SUBROUTINE EVOLV(J1,K2,K1,MM)
      REAL BETAZ0,BETAZ,GAM,KPOD(20),OMEGA(20)
      REAL A(20,5),PHI1(20,5),PHI2(20,5),PSI(20,3000,5),U0(3000,5)
      REAL K2(20),K1(20),TIM,THETA(20,5),APP(20,5),TP(20,5)
      REAL NUI,NUR
      INTEGER J1,MM,N,F,NPART
      COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
      COMMON/BLK5/PSI,U0,PHI1,PHI2,A,THETA,APP,TP
      COMMON/BLK3/TIM,NPART,N,RISE,TAU,BETA1,BETA2,F,NUI,NUR
      DUM1 = 0.
      DUM2 = 0.
      DUM3 = 0.
      DUM4 = 0.
      DO 100 JP =1,NPART
      J = JP + (N-1)*F
      TRISE = 1. - EXP(-FLOAT(J-1)*TAU/RISE)
      BETAZ = BETAZ0*(U0(JP,MM) + OMEGA(1) )/KPOD(1)
      GAM = SQRT( (1.+(GAM0*BETAW)**2)/(1.-BETAZ*BETAZ) )
      PSI(J1,JP,MM)=KPOD(J1)*(PSI(1,JP,MM)+OMEGA(1)*TIM)/KPOD(1)
     1   - OMEGA(J1)*TIM
      DUM1 = DUM1 + COS(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE*GAM0/GAM
      DUM2 = DUM2 + SIN(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE*GAM0/GAM
      DUM3 = DUM3 + COS(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE
100   DUM4 = DUM4 + SIN(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE
      K1(J1) = -NUR*A(J1,MM)/2.- BETA1*DUM2/OMEGA(J1)
      K2(J1) = -NUI/2. + BETA1*DUM1/(A(J1,MM)*OMEGA(J1))
      PHI1(J1,MM) = - BETA2*DUM3/KPOD(J1)**2
      PHI2(J1,MM) = - BETA2*DUM4/KPOD(J1)**2
      RETURN
      END
```

```
LLLLLLLLLL    11111111111111111111111111111111111111111111111111    LLLLLLLLLL
LLLLLLLLLL    Digital Equipment Corporation - VAX/VMS Version V4.6    LLLLLLLLLL
LLLLLLLLLL    11111111111111111111111111111111111111111111111111    LLLLLLLLLL
```

```
M   M   AAA   RRRR    AAA   BBBB   L      EEEEE
MM MM  A   A  R   R  A   A  B   B  L      E
M M M  A   A  R   R  A   A  B   B  L      E
M   M  A   A  RRRR   A   A  BBBB   L      EEEE
M   M  AAAAA  R R    AAAAA  B   B  L      E
M   M  A   A  R  R   A   A  B   B  L      E
M   M  A   A  R   R  A   A  BBBB   LLLLL  EEEEE

H   H  EEEEE  L      W   W   III   GGGG  55555
H   H  E      L      W   W    I   G      5
H   H  E      L      W   W    I   G      555
HHHHH  EEEE   L      W   W    I   G         5
H   H  E      L      W W W    I   G GGG     5
H   H  E      L      WW WW    I   G   G  5   5
H   H  EEEEE  LLLLL  W   W   III   GGG    555

          FFFFF   OOO   RRRR    ;;    1
          F      O   O  R   R   ;;   11
          F      O   O  R   R         1
          FFFF   O   O  RRRR    ;;    1
          F      O   O  R R     ;;    1
       .. F      O   O  R  R     ;    1
       .. F       OOO   R   R   ;    111
```

File _VC$DRB1:[MARABLE.TWSTAG]HELWIG5.FOR;1 (4137,1,0), last revised on 15-FEB-1985 11:34, is a 17 block sequential file owned by UIC [MARABLE]. The records are variable length with a fixed control size of 2 bytes and implied (CR) carriage control. The longest record is 72 bytes.

Job HELWIG5 (1984) queued to LN03_QUE on 21-MAR-1988 13:47 by user MARABLE, UIC [MARABLE], under account 4790 at priority 100, started on printer LTA8: on 21-MAR-1988 13:47 from queue VC_LN03B.


```
LLLLLLLLLL    11111111111111111111111111111111111111111111111111    LLLLLLLLLL
LLLLLLLLLL    Digital Equipment Corporation - VAX/VMS Version V4.6    LLLLLLLLLL
LLLLLLLLLL    11111111111111111111111111111111111111111111111111    LLLLLLLLLL
```

```
C***    CODE TO EVALUATE TEMPORAL EVOLUTION OF THE SPRECTRA
C***    OF UNSTABLE MODES IN A HELICAL WIGGLER FREE ELECTRON LASER
C***    DELETION OF FIRST TRANSIT TIME
        REAL A(5000,25),PSI(25,10000),UOLD(10000),TIME(5000)
        REAL GROWTH(5000,25),DUM(25),DUM1(25),DUM2(25),DUM3(25)
        REAL OMEGA(25),KPOD(25),EWAV(5000),GROE(5000),THETAPP
        REAL KWIGR,KWIGL,NU,NUR,NUI,APRIM(25),PHI1(25),PHI2(25)
        REAL THETAP(5000,25),THETA(25),FILL
        INTEGER F
        CHARACTER*40 XLAB,GLAB,YLAB
        CHARACTER*80 AB1,AB2(25),AB3,AB4
        PARAMETER (PI=3.1415926535)
        PS00(J1,J) = -OMEGA(J1)*FLOAT(J-1)*TAU - THETA(J1)
        TRISE(J)= 1. - EXP(-FLOAT(J-1)*TAU/RISE)
        WRITE(6,10)
        XLAB = ' TIME$'
10      FORMAT( 'INPUT NO. OF PART.,NO. OF ITERAT.,GAM,BETAWIG,KWIGR
        1 ,BUDKER,NWIG,EPS,PHASE,RISE,NPLUS,NMODE,NSEP,REF,F,FILL')
        READ(5,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,PHASE
        1 ,RISE,NPLUS,NMODE,NSEP,REF,F,FILL
20      FORMAT(' INPUT DATA:NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,
        1 EPS,PHASE,RISE,NPLUS,NMODE,NSEP,REF,F,FILL IS:')
        WRITE(6,20)
        WRITE(6,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,PHASE,
        1 RISE,NPLUS,NMODE,NSEP,REF,F,FILL
30      FORMAT(' WAVE AMPLITUDE HELWIG5 W/ R=',F5.3)
        WRITE(AB4,30)REF
        KWIGL = 2.*FLOAT(NWIG)*PI
        BETA0 = SQRT(1. - 1./(GAM0*GAM0))
        BETAZ0 = SQRT(BETA0*BETA0 - BETAW*BETAW)
        NOPT = JIFIX(2.*FLOAT(NWIG)*BETAZ0/(1.-BETAZ0)) +NPLUS
        OMEGA(1) = (FLOAT(NOPT)*PI)/BETAZ0
        KPOD(1) = KWIGL + BETAZ0*OMEGA(1)
        DO 50 J=2,(NMODE-1)/2 + 1
        OMEGA(J) = FLOAT(NOPT+(J-1)*NSEP)*PI/BETAZ0
        KPOD(J) = KWIGL +BETAZ0*OMEGA(J)
        OMEGA(NMODE+2-J) =FLOAT(NOPT-(J-1)*NSEP)*PI/BETAZ0
        KPOD(NMODE+2-J) = KWIGL + BETAZ0*OMEGA(NMODE+2-J)
50      CONTINUE
        WRITE(AB1,104)NPART,F,NTIMES,GAM0,BETAW,NWIG
104     FORMAT('NPART=',I4,2X,'F=',I4,2X,'NTIMES=',I4,2X,'GAM=',F5.3,2X,
        1 'BETAWIG=',F5.3,2X,'NWIG=',I3)
105     FORMAT('N=',I4,3X,'NU=',E10.4,3X,'KPOD=',E10.4
        1,3X,'OMEGA=',E10.4,2X,'NP=',I4)
        WRITE(AB3,106)KWIGL,EPS,FILL,RISE,KWIGR
106     FORMAT('KWIGL=',F8.4,2X,'EPS=',E10.4,2X,'FILL=',F5.3,2X,
        1'RISE=',F5.3,2X,'KWIGR=',F8.4)
        BETA1 = 2.*FILL*NU*KWIGL**2*BETA0*BETAW/(KWIGR**2*BETAZ0**3)
        NUR = (1.-REF)/BETAZ0
        NUI = -4.*FILL*NU*KWIGL**2*(1.-BETAW**2/2.)/(GAM0*BETAZ0*
        1 KWIGR**2*BETAZ0*OMEGA(1))
        BETA2 = 8.*NU*BETA0*KWIGL**2/(BETAZ0*KWIGR**2)
        BETA2 = 0.
        ALPHA2 = .5*BETAW*GAM0/BETAZ0**2
C***    INITIALIZE PHASE AND AMPLITUDE
        TIME(1) = 1.
        GROE(1) = 0.
        TAU = 1./FLOAT(NPART -1)
        TAUT = FLOAT(F)*TAU
        EWAV0 = 0.
C***    INITIALIZE EACH MODE UNDER CONSIDERATION AND FIND
C***    THE WAVE ENERGY DENSITY  IN THE INITIAL SPECTURM
        DO 60 J1=1,NMODE
        APRIM(J1) = 0.
        THETAPP = 0.
        THETAP(1,J1) = 0.
```

```
      THETA(J1) =PHASE
      PHI1(J1) = 0.
      A(1,J1) = EPS
      PHI2(J1) = 0.
      EWAV0 = EWAV0 +(GAM0*KWIGR)**2*((KPOD(J1)-KWIGL)*A(1,J1)/
     1 KWIGL)**2/(4.*NU*(GAM0-1.))
      GROWTH(1,J1) = 0.
      WRITE(AB2(J1),105)JIFIX(OMEGA(J1)*BETAZ0/PI),NU,KPOD(J1)
     1 ,OMEGA(J1),NPLUS
60    CONTINUE
C***  INITIALIZE PHASES FOR THE FIRST MODE
      DO 80 J= 1,NPART
      PSI(1,J) = PS00(1,J) + (KPOD(1)- OMEGA(1))*(1.-FLOAT(J-1)*TAU)
      UOLD(J) = KPOD(1) - OMEGA(1)
80    CONTINUE
C***  EVOLVE PHASES OF PARTICLES FOR TIMES GREATER THAN ONE TRANSIT
C***  TIME
      EWAV(1) = EWAV0
      DO 2000 N = 2,NTIMES
      N1 = N -1
      TIME(N) = FLOAT(N1)*TAUT +1.
C***  INITIALIZE VARAIBLES FOR ENSEMBLE AVERAGE (SUM)
      DO 2080 J1=1,NMODE
      DUM(J1) = 0.
      DUM1(J1) = 0.
      DUM2(J1) = 0.
      DUM3(J1) = 0.
2080  CONTINUE
C*** EVOLVE PARTICLE PHASES FOR PARTICLES ALREADY WITHIN THE RESONATOR
C***  FIRST PERFORM BOOKEEPING TO LOCATE PARTICLES (WITH ENTRANCE TIMES
C***   ASSOCIATED WITH PARTICLES IN THE RESONATOR ON THE PREVIOUS TIME
C***   STEP) IN THE FIRST ELEMENTS OF THE PARTICLE PHASE ARRAY
      DO 2200 J=1, NPART -F
      JP = J + N1*F
      PSI(1,J) = PSI(1,J+F)
      UOLD(J) = UOLD(J+F)
      DO 2210 J1 =2,NMODE
      PSI(J1,J) =KPOD(J1)*(PSI(1,J) +OMEGA(1)*TIME(N)+ THETA(1))/
     1 KPOD(1) - OMEGA(J1)*TIME(N) - THETA(J1)
2210  CONTINUE
      BETAZ = BETAZ0*(UOLD(J) + OMEGA(1) +THETAP(N1,1))/KPOD(1)
      GAM = 1./SQRT(1. -BETAZ*BETAZ -BETAW*BETAW)
      DUM4 = 0.
      DUM5 = 0.
      DO 2350 J1 = 1,NMODE
      DUM4 = DUM4 + KPOD(J1)*(PHI2(J1)*COS(PSI(J1,J))-PHI2(J1)*
     1 SIN(PSI(J1,J)))
      DUM5 = DUM5 +(KPOD(J1)-BETAZ*BETAZ0*(OMEGA(J1)+THETAP(N1,J1)))*
     1 A(N1,J1)*SIN(PSI(J1,J))-BETAZ*BETAZ0*APRIM(J1)*COS(PSI(J1,J))
      DUM(J1) = DUM(J1) + COS(PSI(J1,J))*TAU*TRISE(JP)*GAM0/GAM
      DUM1(J1) = DUM1(J1) + SIN(PSI(J1,J))*TAU*TRISE(JP)*GAM0/GAM
      DUM2(J1) = DUM2(J1) + COS(PSI(J1,J))*TAU*TRISE(JP)
      DUM3(J1) = DUM3(J1) + SIN(PSI(J1,J))*TAU*TRISE(JP)
2350  CONTINUE
C***  EVOLVE THE PHASES OF THE FIRST MODE
      PSI(1,J) = PSI(1,J) + TAUT*UOLD(J)
      UOLD(J) = UOLD(J) + TAUT*((1.-BETAZ*BETAZ)*KPOD(1)*DUM4/
     1(GAM*BETAZ0)+ALPHA2*KPOD(1)*DUM5/(GAM*GAM) -THETAPP)
C***  NOW EVALUATE THE PHASES OF THE OTHER MODES IN TERMS OF
C***  THE FIRST MODE
      DO 2375 J1=2,NMODE
      PSI(J1,J) = KPOD(J1)*(PSI(1,J)+ OMEGA(1)*TIME(N)+ THETA(1))/
     1 KPOD(1) - OMEGA(J1)*TIME(N) - THETA(J1)
2375  CONTINUE
```

```
2200   CONTINUE
C***   NOW COMPLETE THE ENSEMBLE AVERAGE FOR PARTICLE JUST ENTERING
C***   THE RESONATOR
       DO 2600 J =NPART +1-F,NPART
       JP = J + N1*F
       DUM4 = 0.
       DUM5 = 0.
       DO 2450 J1=1,NMODE
       DUM4 = DUM4 + KPOD(J1)*(PHI2(J1)*COS(PS00(J1,JP))-PHI1(J1)*
       1 SIN(PS00(J1,JP)))
       DUM5 = DUM5 +(KPOD(J1)- BETAZ0*BETAZ0*(OMEGA(J1)+THETAP(N1,J1)))
       1 *A(N1,J1)*SIN(PS00(J1,JP))-BETAZ0*BETAZ0*APRIM(J1)*COS(PS00(J1
JP))
       DUM(J1) = DUM(J1) + COS(PS00(J1,JP))*TAU*TRISE(JP)
       DUM1(J1) = DUM1(J1) + SIN(PS00(J1,JP))*TAU*TRISE(JP)
       DUM2(J1) = DUM2(J1) + COS(PS00(J1,JP))*TAU*TRISE(JP)
       DUM3(J1) = DUM3(J1) + SIN(PS00(J1,JP))*TAU*TRISE(JP)
2450   CONTINUE
C***   EVOLVE THE PHASES FOR THE FIRST MODE
       UOLD(J)=KPOD(1)-OMEGA(1)-THETAP(N1,1)+(1.+FLOAT(N1)*TAUT-FLOAT(J
P-1)
       1 *TAU)*((1.-BETAZ0*BETAZ0)*KPOD(1)*DUM4/(GAM0*BETAZ0)
       1 + ALPHA2*KPOD(1)*DUM5/(GAM0*GAM0) - THETAPP )
       PSI(1,J) = PS00(1,JP) +(1.+FLOAT(N1)*TAUT-FLOAT(JP-1)*TAU)*(UOLD
(J)
       1 + KPOD(1) - OMEGA(1) - THETAP(N1,1))/2.
C***   NOW EVALUATE THE PHASES OF THE OTHER MODES FROM THE FIRST MODE
       DO 2460 J1=2,NMODE
       PSI(J1,J) =KPOD(J1)*(PSI(1,J)+OMEGA(1)*TIME(N)+THETA(1))/KPOD(1)
       1 - OMEGA(J1)*TIME(N) - THETA(J1)
2460   CONTINUE
2600   CONTINUE
C***   NOW EVOLVE THE POTENTIALS, GROWTH RATES FOR EACH OF THE MODES
C***    AND EVALUATE THE TOTAL WAVE ENERGY DENSITY
       EWAV(N) = 0.
       DO 2500 J1=1,NMODE
       A(N,J1) = (A(N1,J1)-TAUT*BETA1*DUM1(J1)/OMEGA(J1))/(1.+TAUT*NUR/
2.)
       THETAP(N,J1) = -NUI/2. + BETA1*DUM(J1)/(OMEGA(J1)*A(N,J1))
       PHI1(J1) = - BETA2*DUM2(J1)/(OMEGA(J1)**2)
       PHI2(J1) = - BETA2*DUM3(J1)/(OMEGA(J1)**2)
       GROWTH(N,J1)= 2.*(A(N,J1)-A(N1,J1))/(TAUT*(A(N,J1)+A(N1,J1)))
       THETA(J1) = THETA(J1) + TAUT*THETAP(N,J1)
       EWAV(N) = EWAV(N) +(GAM0*KWIGR)**2*((KPOD(J1)-KWIGL)*A(N,J1)/
       1 KWIGL)**2/(4.*NU*(GAM0-1.))
2500   CONTINUE
       GROE(N) = (EWAV(N) - EWAV0)/(TAUT*EWAV0)
       EWAV0 = EWAV(N)
       THETAPP = 2.*(THETAP(N,1)-THETAP(N1,1))/(THETAP(N,1)+THETAP(N1,1
))
C***   REPEAT FOR NEXT TIME STEP
2000   CONTINUE
       DO 3000 J1=1,NMODE
       YLAB = ' WAVE AMPLITUDE$'
       GLAB = ' WAVE AMPLITUDE VS. TIME MULTI$'
       CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
       CALL PWRIT(500,80,%REF(AB1),70,1,0,0)
       CALL PWRIT(500,48,%REF(AB2(J1)),72,1,0,0)
       CALL PWRIT(500,16, %REF(AB3),75,1,0,0)
       CALL EZXY(TIME,A(1,J1),NTIMES,%REF(AB4))
       YLAB = ' GROWTH RATE $'
       GLAB = ' GROWTH RATE NORMALIZED TO TRANSIT TIME$'
       CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
       CALL EZXY(TIME,GROWTH(1,J1),NTIMES,%REF(GLAB) )
       YLAB = ' FREQUENCY SHIFT$'
       GLAB = ' FREQUENCY SHIFT VS. TIME $'
```

```
      CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
      CALL EZXY(TIME,THETAP(1,J1),NTIMES,%REF(GLAB) )

3000  CONTINUE
      GLAB = 'FIELD ENERGY DENSITY VS. TIME$'
      YLAB = ' ENERGY DENSITY$'
      CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
      CALL EZXY(TIME,EWAV,NTIMES,%REF(GLAB))
      GLAB =' RATE OF CHANGE OF ENERGY$'
      YLAB = 'GROWTH RATE$'
      CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
      CALL EZXY(TIME,GROE,NTIMES,%REF(GLAB))
      STOP
      END
```

```
PPPPPPPPPP    555555555555555555555555555555555555555555555555555555    PPPPPPPPPP
 PPPPPPPPP    Digital Equipment Corporation - VAX/VMS Version V4.6    PPPPPPPPPP
 PPPPPPPPP    555555555555555555555555555555555555555555555555555555    PPPPPPPPPP

          M     M    AAA    RRRR     AAA    BBBB    L       EEEEE
          MM   MM   A   A   R   R   A   A   B   B   L       E
          M M M M   A   A   R   R   A   A   B   B   L       E
          M  M  M   A   A   RRRR    A   A   BBBB    L       EEEE
          M     M   AAAAA   R R     AAAAA   B   B   L       E
          M     M   A   A   R  R    A   A   B   B   L       E
          M     M   A   A   R   R   A   A   BBBB    LLLLL   EEEEE

          M     M    AAA    X   X   W   W   TTTTT   EEEEE   M     M
          MM   MM   A   A   X   X   W   W     T     E       MM   MM
          M M M M   A   A    X X    W   W     T     E       M M M M
          M  M  M   A   A     X     W   W     T     EEEE    M  M  M
          M     M   AAAAA    X X    W W W     T     E       M     M
          M     M   A   A   X   X   WW WW     T     E       M     M
          M     M   A   A   X   X   W   W     T     EEEEE   M     M

                        FFFFF    OOO    RRRR     ;;     1
                        F       O   O   R   R    ;;    11
                        F       O   O   R   R           1
                        FFFF    O   O   RRRR     ;;     1
                        F       O   O   R R      ;;     1
                   ..   F       O   O   R  R      ;     1
                   ..   F        OOO    R   R    ;     111
```

File _VC$DRB1:[MARABLE.FEL]MAXWTEM.FOR;1 (3924,1,0), last revised on 7-OCT-1984 15:18, is a 34 block sequential file owned by UIC [MARABLE]. The records are variable length with a fixed control size of 2 bytes and implied (CR) carriage control. The longest record is 72 bytes.

Job MAXWTEM (2005) queued to LN03_QUE on 21-MAR-1988 14:21 by user MARABLE, UIC [MARABLE], under account 4790 at priority 100, started on printer LTA7: on 21-MAR-1988 14:21 from queue VC_LN03A.


```
PPPPPPPPPP    555555555555555555555555555555555555555555555555555555    PPPPPPPPPP
PPPPPPPPPP    Digital Equipment Corporation - VAX/VMS Version V4.6    PPPPPPPPPP
 PPPPPPPPP    555555555555555555555555555555555555555555555555555555    PPPPPPPPPP
```

```
C MAIN PROGRAM FOR COMPARISON OF DELTA FUNCTION AND STEP FUNCTION
C DISPERSIONS USING MULLER'S METHOD TO DETERMINE THE COMPLEX PART
C OF THE EIGENFREQUENCIES
        INTEGER N,NPREV,MAXIT,I,M,L,NPTS
        DOUBLE PRECISION EP1,EP2,DENS,WIG,LAM,BETA,PERP,BETA1,PERP1,GAM
        DOUBLE PRECISION DUM1,DUM2,DUM3,DUM4,X(9,50),X2(9,50)
        DOUBLE PRECISION TS,TD,LINC,LINC2,LINC3,Y1(8),OMEG
        DOUBLE PRECISION ALP1,ALP2,DEL
        CHARACTER*40 XLAB
        DOUBLE COMPLEX ZEROS(6),ROOT,PREVRT(20)
        EXTERNAL FN1,FN2
        LOGICAL FNREAL,FSTG
C       FSTG IS A LOGICAL VARIABLE: IF TRUE THE FIRST GUESS TO THE ROOTS
C        OF THE DISPERSION RELATION ARE THE ROOTS FROM THE PREVIOUS
C        INCREMENT OF K. IF FALSE THE FIRST GUESS IS FROM PREVIOUS DATA
C        AT THE SAME K FROM THE FILE DATINI.
        COMMON DENS,WIG,LAM,BETA,BETA1,PERP,PERP1,GAM,ALP1,ALP2,DEL
        OPEN (UNIT =1, NAME='OUTFILE' ,STATUS ='NEW')
        OPEN (UNIT =2, NAME='DATINI' ,STATUS ='OLD')
        FNREAL= .FALSE.
        MAXIT = 1000
        EP1 = 1.D-15
        EP2 = 1.D-15
        WRITE(6,1)
        WRITE(1,1)
1       FORMAT(' INPUT FOLLOWING DATA: DENS,WIG,LAM,LINC,LINC2,LINC3,BET
A
        1 ,PERP,DEL,NPTS,FIRSTG  ')
        READ(5,*)DENS,WIG,LAM,LINC,LINC2,LINC3,BETA,PERP,DEL,NPTS,FSTG
        READ(2,*)((X2(K,J),K=1,9),J=1,NPTS)
        WRITE(1,*) DENS,WIG,LAM,LINC,LINC2,LINC3,BETA,PERP,DEL,NPTS,FSTG
        WRITE(6,*) DENS,WIG,LAM,LINC,LINC2,LINC3,BETA,PERP,DEL,NPTS,FSTG
        GAMBAR = DSQRT(1. + BETA*BETA + PERP +WIG)
        BETA = BETA/GAMBAR
        PERP = PERP/(GAMBAR*GAMBAR)
        WIG = WIG/(GAMBAR*GAMBAR)
        DENS = DENS/GAMBAR
        DEL = DEL/GAMBAR
        BETA1 = BETA/DSQRT(1.-PERP)
        PERP1 = PERP/(1.-PERP)
        GAM = 1./DSQRT(1.-PERP)
        TD = PERP*GAMBAR*(1. - .5*WIG)
        TS = .5*PERP1*GAMBAR*(1. - .5*WIG - PERP1/3.)/GAM
        WRITE(6,3) TD,TS
        WRITE(1,3) TD,TS
3       FORMAT(' DELTA TEMP = ',D13.4,' STEP TEMP = ',D13.4)
        WRITE(6,2)
        WRITE(1,2)
2       FORMAT(' LAM          2nd ROOT      DLD     DTP1D     DTM1D    DLS
        1   DTP1S   DTM1S')
        DO 1000 L = 1,NPTS
        IF( L .LE. 20) LAM = LAM + LINC
        IF( L .GT. 20 .AND. L .LE. 30) LAM = LAM + LINC2
        IF( L .GT. 30) LAM = LAM + LINC3
        ALP1 = DEL*LAM*(1.- BETA*BETA)
        ALP2 = 1. - PERP1/4. -BETA1*BETA1*(1. - 3.*PERP1/4.)
        X(1,L) = LAM
        N = 6
        NPREV = 0
        ZEROS(1) = DCMPLX(X2(3,L),X2(2,L))
        ZEROS(2) = (0.,0.)
        ZEROS(3) = (0.,0.)
        ZEROS(4) = (0.,0.)
        ZEROS(5) = (0.,0.)
        ZEROS(6) = (0.,0.)
        IF (L .GT. 1 .AND. FSTG) THEN
```

```
              ZEROS(1) = PREVRT(1)
              ZEROS(2) = PREVRT(2)
              ZEROS(3) = PREVRT(3)
              ZEROS(4) = PREVRT(4)
              ZEROS(5) = PREVRT(5)
              ZEROS(6) = PREVRT(6)
              END IF
              CALL MULLER(FN1,FNREAL,ZEROS,N,NPREV,MAXIT,EP1,EP2,1)
      C       FIND EIGENVALUES WITH THE LARGEST GROWTH RATES
              CALL SORT1(N,ZEROS,DUM1,DUM2,DUM3,DUM4)
              Y1(1) = DUM3
              Y1(2) = DUM4
              X(2,L) = DUM1
              X(3,L) = DUM2
              PREVRT(1) = ZEROS(1)
              PREVRT(2) = ZEROS(2)
              PREVRT(3) = ZEROS(3)
              PREVRT(4) = ZEROS(4)
              PREVRT(5) = ZEROS(5)
              PREVRT(6) = ZEROS(6)
      C  ****** X2 IS THE LARGEST GROWTH RATE IN THEN DATA SET FOR FULL DELTA
      C ******  X3 IS THE REAL FREQUENCY CORESPONDING TO THE ABOVE
      C *******  Y1 IS THE SECOND LARGEST DISTINCT IMAGINARY FREQUENCY
      C ******* Y2 IS THE REAL FREQUENC CORRESPONDING TO THE ABOVE
              OMEG = DUM2 - BETA*LAM
              Y1(3) = LAM**2*(OMEG**2 - DENS*(1.-BETA**2))
              Y1(4) = DUM2**2 -(LAM +1.)**2 - DENS*(1.-PERP/2.)
              Y1(5) = DUM2**2 -(LAM-1.)**2 - DENS*(1. - PERP/2.)
              NPREV = 0
              ZEROS(1) = DCMPLX(X2(5,L),X2(4,L))
              ZEROS(2) = (0.,0.)
              ZEROS(3) = (0.,0.)
              ZEROS(4) = (0.,0.)
              ZEROS(5) = (0.,0.)
              ZEROS(6) = (0.,0.)
              IF(L .GT. 1 .AND. FSTG) THEN
              ZEROS(1) = PREVRT(7)
              ZEROS(2) = PREVRT(8)
              ZEROS(3) = PREVRT(9)
              ZEROS(4) = PREVRT(10)
              ZEROS(5) = PREVRT(11)
              ZEROS(6) = PREVRT(12)
              END IF
              CALL MULLER(FN2,FNREAL,ZEROS,N,NPREV,MAXIT,EP1,EP2,1)
              CALL SORT1(N,ZEROS,DUM1,DUM2,DUM3,DUM4)
              X(4,L) = DUM1
              X(5,L) = DUM2
              PREVRT(7) = ZEROS(1)
              PREVRT(8) = ZEROS(2)
              PREVRT(9) = ZEROS(3)
              PREVRT(10) = ZEROS(4)
              PREVRT(11) = ZEROS(5)
              PREVRT(12) = ZEROS(6)
      C ****  X4 IS THE LARGEST GROWTH RATE IN THE DATA SET FOR FULL STEP
      C ****  X5 IS THE REAL FREQUENCY CORESPONDING TO THE ABOVE
              OMEG = DUM2 - BETA1*LAM*(1. - PERP1/4.)
              Y1(6) = DENS*GAM*((1.-BETA1**2)-PERP1*(1.-3.*BETA1**2)/4.)
              Y1(6) = LAM**2*(OMEG**2 - Y1(6))
              Y1(7) = DUM2**2-(LAM+1.)**2-DENS*(1.-PERP1/2.+(PERP1/2.)**2)*GAM
              Y1(8) = DUM2**2-(LAM-1.)**2-DENS*(1.-PERP1/2.+(PERP1/2.)**2)*GAM
              N = 4
              NPREV = 0
              ZEROS(1) = DCMPLX(X2(7,L),X2(6,L))
              ZEROS(2) = (0.,0.)
              ZEROS(3) = (0.,0.)
              ZEROS(4) = (0.,0.)
```

```
         IF (L .GT. 1 .AND. FSTG) THEN
         ZEROS(1) = PREVRT(13)
         ZEROS(2) = PREVRT(14)
         ZEROS(3) = PREVRT(15)
         ZEROS(4) = PREVRT(16)
         END IF
         CALL MULLER(FN1,FNREAL,ZEROS,N,NPREV,MAXIT,EP1,EP2,2)
         CALL SORT1(N,ZEROS,DUM1,DUM2,DUM3,DUM4)
         X(6,L) = DUM1
         X(7,L) = BETA*LAM
         PREVRT(13) = ZEROS(1)
         PREVRT(14) = ZEROS(2)
         PREVRT(15) = ZEROS(3)
         PREVRT(16) = ZEROS(4)
C ***** X6 IS THE LARGEST GROWTH RATE FOR THE REFERENCE DELTA FUNCTION
C ***** X7 IS AN APPROXIMATION TO THE REAL FREQUENCY FOR SMALL DENSITIES
         NPREV = 0
         ZEROS(1) = DCMPLX(X2(9,L),X2(8,L))
         ZEROS(2) = (0.,0.)
         ZEROS(3) = (0.,0.)
         ZEROS(4) = (0.,0.)
         IF (L .GT. 1 .AND. FSTG) THEN
         ZEROS(1) = PREVRT(17)
         ZEROS(2) = PREVRT(18)
         ZEROS(3) = PREVRT(19)
         ZEROS(4) = PREVRT(20)
         END IF
         CALL MULLER(FN2,FNREAL,ZEROS,N,NPREV,MAXIT,EP1,EP2,2)
         CALL SORT1(N,ZEROS,DUM1,DUM2,DUM3,DUM4)
         X(8,L) = DUM1
         X(9,L) = BETA1*LAM
         PREVRT(17) = ZEROS(1)
         PREVRT(18) = ZEROS(2)
         PREVRT(19) = ZEROS(3)
         PREVRT(20) = ZEROS(4)
C ****** X8 IS THE LARGEST GROWTH RATE FOR THE REFERENCE STEP FUNCTION
C ***** X9 IS AN APPROXIMATION OF THE REAL FREQUENCY FOR SMALL DENSITIES
         WRITE(1,5) LAM,(Y1(K),K=1,4)
         WRITE(1,5)(Y1(K),K=5,8)
         WRITE(6,5) LAM,(Y1(K),K=1,4)
         WRITE(6,5)(Y1(K),K=5,8)
1000     CONTINUE
         WRITE(1,6)
         WRITE(6,6)
6        FORMAT('   LAM        FULL DELTA       FULL STEP        REF DELTA
         1      REF STEP ')
         DO 2000 J = 1,NPTS
         WRITE(1,5)(X(K,J),K=1,5)
         WRITE(1,5)(X(K,J),K=6,9)
         WRITE(1,*)
         WRITE(6,5)(X(K,J),K=1,5)
         WRITE(6,5)(X(K,J),K=6,9)
         WRITE(2,*)(X(K,J),K=1,9)
2000     CONTINUE
C ****   PLOT FULL DELTA vs. REF. DELTA
         XLAB = ' COMPARE FULL DELTA AND REF. DELTA'
         CALL QPICTR(X,18,NPTS,QY(3,11),QX(1),QMOVE(00),QXLAB(XLAB),QLABE
L(14))


C   ***  PLOT FULL DELTA vs. FULL STEP
         XLAB = ' COMPARE FULL DELTA AND FULL STEP'
         CALL QPICTR(X,18,NPTS,QY(3,7),QX(1),QMOVE(00),QXLAB(XLAB),QLABEL
(14))
```

```
C ****   PLOT REF. DELTA vs. REF STEP
         XLAB = ' COMPARE REF. DELTA AND REF. STEP'
         CALL QPICTR(X,18,NPTS,QY(11,15),QX(1),QMOVE(00),QXLAB(XLAB),QLAB
EL(14))


C ****   PLOT FULL STEP vs. REF STEP
         XLAB = ' COMPARE FULL STEP AND REF. STEP'
         CALL QPICTR(X,18,NPTS,QY(7,15),QX(1),QMOVE(00),QXLAB(XLAB),QLABE
(14))

C *****  PLOT REAL FREQUENCIES FOR FULL DELTA AND STEP FUNCTION EQUILBRIA
         XLAB = ' COMPARE REAL FREQ. FOR DELTA & STEP'
         CALL QPICTR(X,18,NPTS,QY(5,9,13,17),QX(1),QMOVE(00),QXLAB(XLAB))



5        FORMAT(5D14.4)

         STOP
         END
         SUBROUTINE FN1(Z,FZ)
         DOUBLE COMPLEX Z,FZ(2),DLD,DTP1D,DTM1D,CHIBD,CHIAD,OMEG
         DOUBLE COMPLEX ZF,ZFP
         DOUBLE PRECISION DENS,WIG,LAM,BETA,BETA1,PERP,PERP1,GAM
         DOUBLE PRECISION ALP1,ALP2,DEL
         COMMON DENS,WIG,LAM,BETA,BETA1,PERP,PERP1,GAM,ALP1,ALP2,DEL
         OMEG = (Z -BETA*LAM)/ALP1
         CALL ZETA(OMEG,ZF,ZFP)
         ZF = OMEG*OMEG*ZFP/(DEL*DEL*(1.-BETA*BETA))
         DLD = LAM*LAM*OMEG*OMEG - DENS*ZF
         DTP1D = Z*Z -(LAM+1.)*(LAM+1.)-DENS*(1.-PERP/2.)
         DTM1D = Z*Z -(LAM-1.)*(LAM-1.)-DENS*(1.-PERP/2.)
         CHIBD = DENS*(BETA*DEL*OMEG*(1.-3.*PERP/2.)+PERP/2.-1.)*ZF
         CHIAD = DENS*OMEG*OMEG*(1.-PERP/2.) - DENS*((1.-PERP/2.)*
         1 (1.-PERP/2.)-2.*BETA*DEL*OMEG*(1.-PERP/2.)*(1.-3*PERP/2.))*ZF
         FZ(1) =DLD*DTP1D*DTM1D + (WIG/2.)*(DTP1D +DTM1D)*(LAM*LAM*CHIAD
         1  -DENS*DENS*(DEL*DEL*BETA*BETA*(1.-3.*PERP/2.)*(1.-3.*PERP/2.)
         1 *ZF*ZF +(1.- PERP/2.)*ZF))
         FZ(2) = DENS*ZF*DLD*DTM1D/(LAM*LAM) - (WIG/2.)*CHIBD*CHIBD

         RETURN
         END
         SUBROUTINE FN2(Z,FZ)
         DOUBLE COMPLEX Z,FZ(2),DLS,DTP1S,DTM1S,CHIBS,CHIAS,OMEG
         DOUBLE COMPLEX ZF,ZFP
         DOUBLE PRECISION DENS,WIG,LAM,BETA,BETA1,PERP,PERP1,GAM
         DOUBLE PRECISION ALP1,ALP2,DEL
         COMMON DENS,WIG,LAM,BETA,BETA1,PERP,PERP1,GAM,ALP1,ALP2,DEL
         OMEG = (Z -BETA1*LAM*(1.-PERP1/4.))/(DEL*GAM*LAM*ALP2)
         CALL ZETA(OMEG,ZF,ZFP)
         ZF = OMEG*OMEG*ZFP/(DEL*DEL*ALP2)
         DLS = LAM*LAM*OMEG*OMEG - DENS*ZF/GAM
         DTP1S= Z*Z-(LAM+1.)*(LAM+1.)-DENS*GAM*(1.-PERP1/2.+PERP1*PERP1/4
)
         DTM1S= Z*Z-(LAM-1.)*(LAM-1.)-DENS*GAM*(1.-PERP1/2.+PERP1*PERP1/4
.)
         CHIBS =DENS*(BETA1*DEL*GAM*OMEG*(1.-3.*PERP1/2.)+PERP1/2.-1.)*ZF
         CHIAS =DENS*GAM*GAM*GAM*(1.-9.*PERP1/4.)*OMEG*OMEG -DENS*GAM*(
         1 (1.-PERP1/2.)*(1.-PERP1/2.)-2.*BETA1*DEL*GAM*OMEG*(1.-PERP1/2.
)
         1 *(1.-3.*PERP1/2.))*ZF
         FZ(1) = DLS*DTP1S*DTM1S+(WIG/2.)*(DTP1S+DTM1S)*(LAM*LAM*CHIAS
         1 -DENS*DENS*GAM*GAM*(BETA1*BETA1*DEL*DEL*(1.-3.*PERP1/2.)*
         1 (1.-3.*PERP1/2.)*ZF*ZF + (1.-9.*PERP1/4.)*ZF))
         FZ(2) = DENS*ZF*DLS*DTM1S/(LAM*LAM*GAM) -(WIG/2.)*CHIBS*CHIBS
```

```
      RETURN
      END
      SUBROUTINE MULLER(FN,FNREAL,ZEROS,N,NPREV,MAXIT,EP1,EP2,M)
C DETERMINES UP TO N ZEROS OF THE FUNCTION SPECIFIED BY FN USING
C QUADRATIC INTERPOLATION, i.e. MULLER'S MEHTOD
      EXTERNAL FN1,FN2
      LOGICAL FNREAL
      INTEGER MAXIT,N,NPREV,KOUNT,L,M
      DOUBLE PRECISION EP1,EP2,EPS1,EPS2
      DOUBLE COMPLEX ZEROS(N),C,DEN,DIVDF1,DIVDF2,DVDF1P,FZR(2),FZRDFL
     1,FZRPRV,H,ZERO,SQR,HPREV,FN
C *********************  INPUT    *******************************
C   FN NAME OF SUBROUTINE, OF THE FORM FN(X,FX) WHICH FOR GIVEN X
C   RETURNS F(X), THIS MUST APPEAR IN AN EXTERNAL STATEMENT IN MAIN
C   CALLING PROGRAM.
C   FNREAL IS A LOGICAL VARIABLE, IF .TRUE. ALL APPROX. ARE TAKEN
C   TO BE REAL , ALLOWING THIS ROUTINE TO BE USED EVEN IF F(X) IS ONLY
C   DEFINED FOR REAL X.
C   ZEORS(1).... ZEROS(NPREV) CONTAINS PREVIOUSLY FOUND ZEROS OF THE
C   FUNCTION, PROVIDED NPREV .GT. 0
C    ZEROS(NPREV+1).... ZEROS(N) CONTAINS FIRS GUESS FOR THE ZEROS
C     TO BE FOUND
C    MAXIT IS THE MAXIMUM NUMBER OF FUNCTION EVALUATIONS ALLOWED @ ZERO.
C   EP1 ITERATION IS STOPPED IF ABS(H) .LT. EP1*ABS(ZR), WITH
C   H EQUAL TO THE LATEST CHANGE IN THE ZERO ESTIMATE
C   EP2 ALTHOUT THE EP1 CRITERION IS NOT MET, ITERATION IS STOPPED IF
C               ABS(F(ZERO)) .LT. EP2
C  N IS THE TOTAL NUMBER OF ZEROS TO BE FOUND
C   NPREV IS THE NUMBER OF ZEROS FOUND PREVIOUSLY
C********************    OUTPUT    ***************************
C   ZEROS(NPREV +1) ... ZEROS(N)  APPROXIMATIONS TO ZEROS
C            INITIALIZTION
      EPS1 = DMAX1(EP1,1.D-12)
      EPS2 = DMAX1(EP2,1.D-20)
c
      DO 500 I=NPREV +1,N
      KOUNT = 0
C    COMPUTE FIRST THREE ESTIMATES FOR ZERO AS ...
C     ZEROS(I)+.5,  ZEROS(I) - .5,    ZEROS(I)
401   ZERO = ZEROS(I)
      H = .5
      CALL DFLATE(FN,ZERO+.5,I,KOUNT,FZR,DVDF1P,ZEROS,L,M)
      IF(L .NE. 0) GO TO 401
      CALL DFLATE(FN,ZERO-.5,I,KOUNT,FZR,FZRPRV,ZEROS,L,M)
      IF(L .NE. 0) GO TO 401
      HPREV = -1.
      DVDF1P = (FZRPRV - DVDF1P)/HPREV
      CALL DFLATE(FN,ZERO,I,KOUNT,FZR,FZRDFL,ZEROS,L,M)
      IF(L .NE. 0) GO TO 401
C    DO WHILE KOUNT .LE. MAXIT OF H IS RELATIVELY BIG
C     OR FZR = F(ZERO) IS NOT SMALL
C      OR FZRDFL = FDFLATED(ZERO) IS NOT SMALL OR NOT MUCH
C     BIGGER THAN ITS PREVIOUS VALUE FZRPRV.
440   DIVDF1 = (FZRDFL - FZRPRV)/H
      DIVDF2 = (DIVDF1 - DVDF1P)/(H + HPREV)
      HPREV = H
      DVDF1P = DIVDF1
      C = DIVDF1 + H*DIVDF2
      SQR = C*C - 4.*FZRDFL*DIVDF2
      IF (FNREAL .AND. DREAL(SQR) .LT. 0.) SQR = (0.,0.)
      SQR = CDSQRT(SQR)
      IF (DREAL(C)*DREAL(SQR)+DIMAG(C)*DIMAG(SQR) .LT. 0.) THEN
        DEN = C - SQR
      ELSE
        DEN = C + SQR
      END IF
```

```
              IF(CDABS(DEN) .LE. 0.) DEN =(1.,0.)
              H = -2.*FZRDFL/DEN
              FZRPRV = FZRDFL
              ZERO = ZERO + H
              IF(KOUNT .GT. MAXIT) GO TO 499
 470          CALL DFLATE(FN,ZERO,I,KOUNT,FZR,FZRDFL,ZEROS,L,M)
              IF(L .NE. 0) GO TO 401
 C            CHECK FOR CONVERGENCE
              IF(CDABS(H) .LT. EPS1*CDABS(ZERO)) GO TO 499
              IF(DMAX1(CDABS(FZR(M)),CDABS(FZRDFL)) .LT. EPS2) GO TO 499
 C            CHECK FOR DIVERGENCE
              IF(CDABS(FZRDFL) .GE. 10.*CDABS(FZRPRV)) THEN
              H = H/2.
              ZERO = ZERO - H
              GO TO 470
              ELSE
              GO TO 440
              END IF
 499          ZEROS(I) = ZERO
 500          CONTINUE
              RETURN
              END
              SUBROUTINE DFLATE(FN,ZERO,I,KOUNT,FZERO,FZRDFL,ZEROS,L,M)
 C            TO BE CALLED BY MULLER
              INTEGER I,KOUNT,J,L,M
              DOUBLE COMPLEX FZERO(2),FZRDFL,ZERO,ZEROS(8),DEN
              L= 0
              KOUNT = KOUNT + 1
              CALL FN(ZERO,FZERO)
              FZRDFL = FZERO(M)
              IF(I .LT. 2) RETURN
              DO 410 J=2,I
              DEN = ZERO - ZEROS(J-1)
              IF(CDABS(DEN) .EQ. 0.) THEN
              ZEROS(I) = ZERO * 1.001
              L= 1
              RETURN
              ELSE
              FZRDFL = FZRDFL/DEN
              END IF
 410          CONTINUE
              RETURN
              END
              SUBROUTINE SORT1(N,ZEROS,DUM1,DUM2,DUM3,DUM4)
              DOUBLE COMPLEX ZEROS(N)
              DOUBLE PRECISION DUM1,DUM2,DUM3,DUM4
              INTEGER I,J,K,N
 C            FIND THE LARGEST GROWTH RATE IN THE DATA SET ZEROS(N)
              DUM1 = DIMAG(ZEROS(1))
              J = 1
              DO 100 I = 2,N
              IF (DUM1 .GT. DIMAG(ZEROS(I))) THEN
              DUM1 = DUM1
              J = J
              ELSE
              DUM1 = DIMAG(ZEROS(I))
              J = I
              END IF
 100          CONTINUE
              DUM2 = DREAL(ZEROS(J))
              IF(DUM1 .LE. 0.) DUM1 = 0.
 C            DUM1  IS THE LARGEST GROWTH RATE IN THE DATA SET
 C            DUM2 IS THE REAL FREQUENCY CORRESPONDING TO THE MAX GROWTH
 C        NEXT FIND THE NEXT LARGEST GROWTH RATE THAT IS NOT EQUAL IN
 C        MAGNITUDE TO THE FIRST GROWTH RATE
              IF(J .EQ. 1) J = 3
```

```
      M = J - 1
      DUM3 = DIMAG(ZEROS(M))
      DO 300 I=1,N
      IF( DIMAG(ZEROS(I)) .GE. DUM3 .AND. DIMAG(ZEROS(I)) .NE.
      1 DUM1) THEN
      DUM3 = DIMAG(ZEROS(I))
      K = I
      END IF
300   CONTINUE
      DUM4 = DREAL(ZEROS(K))
C     DUM3 IS THE SECOND LARGEST DISTINCT GROWTH RATE
C     DUM4 IS THE REAL FREQUENCY CORRESPONDING TO THE ABOVE
      RETURN
      END
      subroutine zeta(arg1,arg2,arg3)
c     plasma dispersion funcion. see fried and conte for definition.
c     received from d.g. swanson, 8/12/64 cal teck. tested by martha
c     pennell,rle, mit. shifted into fortran iv by m. liedberman. 1/67
c     a.b. lagdon, 6/69- rewrote extensively. is now intelligible and
c     much faster. accuracy same for small arguments. derivative now
c     works for large arguments.
c
c
c     arg1 = argument of z function
c     arg2 = value of z function
c     arg3 = value of derivative of z function
c
c
      double complex arg1,arg2,arg3
      double complex z,a1,a2,a3,b1,b2,b3,z1,zsq,au1,au5
      double complex bb
      double precision x,y,yabs,aa,daa,error,u1,u2,u3,u4,u5
      double precision s1,s2,fn
      equivalence (bb,rbb)
      z = arg1
      x = dreal(z)
      y = dimag(z)
      yabs = dabs(y)
      if(yabs .lt. 1.) goto 10
c
c
c     continued fraction method
      z = dcmplx(x,yabs)
      aa = 0.
      daa = 1.5
      bb = 1.5 - z*z
      a1 = 0.
      a2 = -1.
      b1 = 1.
      b2 = bb
      au1 = a2/b2
3     aa = aa - daa
      daa = daa + 2.
      rbb = rbb + 2.
      a3 = a2*bb + a1*aa
      b3 = b2*bb + b1*aa
      z1 = a3/b3
      au5 = z1 - au1
      if(dabs(dreal(au5))+dabs(dimag(au5)).lt. 1.d-7) goto 5
      a1 = a2
      b1 = b2
      a2 = a3
      b2 = b3
      au1 = z1
      n = n + 1
      goto 3
```

```
 5    if(y) 52,51,51
52    z = dcmplx(x,y)
      if(y*y-x*x .ge. -85.0) then
      z1 = dconjg(z1) -(0.,7.08981540362206)*z*cdexp(-z*z)
      else
      z1 = dconjg(z1)
      end if
51    arg3 = z1
      arg2 = -(1.+.5*z1)/z
      return
c
10    if(abs(x) .lt. 4.) goto 20
c
c
c     asymptotic series method
      zsq = z*z
      z1 = (0.,1.77245385090551)*cdexp(-zsq)
      n = 1
      au1 = 0.5/zsq
      au5 = 1.0/z
11    au5 = au5*au1*dflotj(n)
      z1 = z1 - au5
      error = dabs(dreal(au5))+dabs(dimag(au5))
      n = n + 2
      if(error .gt. 1.d-7) goto 11
      arg3 = -2.*z*z1
      arg2 = z1 - 1.0/z
      return
c
c
c     power series method. needs double precision u's and s's on some
c     computers. change abs to dabs, real to dreal aimag to dimag also
20    u1 = -2.0*(x*x - y*y)
      u2 = -4.0*x*y
      error = 1.d-7/(dabs(dreal(z))+dabs(dimag(z))+1.d-7)
      s1 = 1.0
      s2 = 0.0
      n = 3
      u5 = dflotj(n)
      u3 = u1/u5
      u4 = u2/u5
21    s1 = s1 + u3
      s2 = s2 + u4
      if(dabs(u3)+dabs(u4) .lt. error) goto 25
      n = n + 2
      fn = dflotj(n)
      u5 = (u3*u1-u4*u2)/fn
      u4 = (u4*u1+u3*u2)/fn
      u3 = u5
      go to 21
25    z1 = (0.,1.77245385090551)*cdexp(-z*z)-2.*z*dcmplx(s1,s2)
      arg3 = -2.*(1.+z1*z)
      arg2 = z1
      return
      end
```

```
KKKKKKKKKK    00000000000000000000000000000000000000000000000000    KKKKKKKKKK
KKKKKKKKKK    Digital Equipment Corporation - VAX/VMS Version V4.6    KKKKKKKKKK
KKKKKKKKKK    00000000000000000000000000000000000000000000000000    KKKKKKKKKK

            M   M    AAA    RRRR     AAA    BBBB    L       EEEEE
            MM MM   A   A   R   R   A   A   B   B   L       E
            M M M   A   A   R   R   A   A   B   B   L       E
            M   M   A   A   RRRR    A   A   BBBB    L       EEEE
            M   M   AAAAA   R R     AAAAA   B   B   L       E
            M   M   A   A   R  R    A   A   B   B   L       E
            M   M   A   A   R   R   A   A   BBBB    LLLLL   EEEEE

        MM        MM  UU        UU  LL            TTTTTTTTTT    IIIIII
        MM        MM  UU        UU  LL            TTTTTTTTTT    IIIIII
        MMMM    MMMM  UU        UU  LL                TT          II
        MMMM    MMMM  UU        UU  LL                TT          II
        MM  MM    MM  UU        UU  LL                TT          II
        MM  MM    MM  UU        UU  LL                TT          II
        MM        MM  UU        UU  LL                TT          II
        MM        MM  UU        UU  LL                TT          II
        MM        MM  UU        UU  LL                TT          II
        MM        MM  UU        UU  LL                TT          II
        MM        MM  UU        UU  LL                TT          II
        MM        MM  UU        UU  LL                TT          II
        MM        MM  UUUUUUUUUU    LLLLLLLLLL        TT        IIIIII
        MM        MM  UUUUUUUUUU    LLLLLLLLLL        TT        IIIIII

             FFFFFFFFFF    OOOOOO      RRRRRRRR          ;;;;        11
             FFFFFFFFFF    OOOOOO      RRRRRRRR          ;;;;        11
             FF          OO      OO    RR      RR        ;;;;      1111
             FF          OO      OO    RR      RR        ;;;;      1111
             FF          OO      OO    RR      RR                    11
             FF          OO      OO    RR      RR                    11
             FFFFFFFF    OO      OO    RRRRRRRR          ;;;;        11
             FFFFFFFF    OO      OO    RRRRRRRR          ;;;;        11
             FF          OO      OO    RR  RR            ;;;;        11
             FF          OO      OO    RR  RR            ;;;;        11
      ....   FF          OO      OO    RR    RR            ;;        11
      ....   FF          OO      OO    RR    RR            ;;        11
      ....   FF            OOOOOO      RR      RR        ;;        111111
      ....   FF            OOOOOO      RR      RR        ;;        111111

File _VC$DRB1:[MARABLE.TWSTAG]MULTI.FOR;1 (4141,1,0), last revised on
 3-APR-1985 14:56, is a 14 block sequential file owned by UIC [MARABLE].  The
 ecords are variable length with a fixed control size of 2 bytes and implied
(CR) carriage control.  The longest record is 69 bytes.

 ob MULTI (1983) queued to LN03_QUE on 21-MAR-1988 13:44 by user MARABLE, UIC
 MARABLE], under account 4790 at priority 100, started on printer LTA8: on
21-MAR-1988 13:45 from queue VC_LN03B.

 KKKKKKKKK    00000000000000000000000000000000000000000000000000    KKKKKKKKKK
KKKKKKKKKK    Digital Equipment Corporation - VAX/VMS Version V4.6    KKKKKKKKKK
 KKKKKKKKK    00000000000000000000000000000000000000000000000000    KKKKKKKKKK
```

```
C***    TEST PROGRAM FOR MODEL EQUATIONS IN HIGH POWER
C***    FEL PROBLEM.
C***    REVISION 12/3 TO CORRECT BOUNDARY CONDITIONS
C***    REVISION 1/14 TO INCLUDE MULTIPLE MODES
C***    REVISION 1/15 TO INCLUDE GROWTH OF TOTAL WAVE ENERGY
C***    REVISION 1/16 TO INCLUDE PHENOMENOLOGICAL DAMPING
        REAL A(3000,50),PSIM(4000,50),PSI0(4000,50),EPS,BETA1,TAU,DUM(50
)
        REAL TIME(3000),UOLD(4000,50),UNEW(4000,50),GAM(3000)
        REAL GROWTH(4000,50),KWIGR,KWIGL,NU,RISE,DUM1(50),OMEGA(50),KPOD
(50)
        REAL ARPRIM(50),AIMPRIM(50),ARN(50),ARN1(50),AIMN(50),AIMN1(50)
        REAL GROE(3000),EWAV(3000)
        INTEGER J,NPART,NTIMES,N,F,NMODE,NSEP
        PARAMETER (PI=3.141592653589)
        CHARACTER*40 GLAB,XLAB,YLAB
        CHARACTER*80 AB1,AB2(50),AB3,AB4
        PS00(M,J) = -OMEGA(M)*FLOAT(J)*TAU
1       FORMAT( 'WAVE AMPLITUDE MULTI W/ R=',F5.3)
        XLAB =' TIME$'
        YLAB = ' WAVE AMPLITUDE$'
        WRITE(6,101)
101     FORMAT(' INPUT NO. OF PART.,NO. OF INTERAT.,GAM,BETAWIG,
        1 KWIGr,BUDKER,NWIG,EPS,F,RISE,NPLUS,NMODE,NSEP,REF ')
        READ(5,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,F,RISE,NPLUS
        1 ,NMODE,NSEP,REF
        WRITE(6,103)
103     FORMAT(' INPUT DATA:NPART,NTIMES,GAM,BETAW,KWIGR,NU,NWIG
        1,EPS,F,RISE,NPLUS,NMODE,NSEP,REF IS:')
        WRITE(6,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,F,RISE,NPLUS
        1 ,NMODE,NSEP,REF
        KWIGL = 2.*FLOAT(NWIG)*PI
        BETA0 = SQRT(1. - 1./(GAM0*GAM0))
        BETAZ0 = SQRT(BETA0*BETA0 - BETAW*BETAW/2.)
        NOPT = JIFIX(2.*FLOAT(NWIG)*BETAZ0/(1.-BETAZ0)) +NPLUS
        OMEGA(1) = (FLOAT(NOPT)*PI)/BETAZ0
        KPOD(1) = KWIGL + BETAZ0*OMEGA(1)
        DO 50 J=2,(NMODE-1)/2 + 1
        OMEGA(J) = FLOAT(NOPT+(J-1)*NSEP)*PI/BETAZ0
        KPOD(J) = KWIGL +BETAZ0*OMEGA(J)
        OMEGA(NMODE+2-J) =FLOAT(NOPT-(J-1)*NSEP)*PI/BETAZ0
        KPOD(NMODE+2-J) = KWIGL + BETAZ0*OMEGA(NMODE+2-J)
50      CONTINUE
        WRITE(AB4,1)REF
        WRITE(AB1,104)NPART,NTIMES,GAM0,BETAW,NWIG
104     FORMAT('NPART=',I4,2X,'NTIMES=',I4,2X,'GAM=',F7.5,2X,
        1 'BETAWIG=',F8.4,2X,'NWIG=',I3)
105     FORMAT('N=',I4,3X,'NU=',E10.4,3X,'KPOD=',E10.4
        1,3X,'OMEGA=',E10.4,2X,'NP=',I4)
        WRITE(AB3,106)KWIGL,EPS,F,RISE,KWIGR
106     FORMAT('KWIGL=',F8.4,2X,'EPS=',E10.4,2X,'F=',I3,2X,
        1'RISE=',F8.4,2X,'KWIGR=',F8.4)
        BETA1 = 4.*NU*BETAW*(KWIGL)**2/(BETAZ0*BETA0*GAM0
        1 *(KWIGR*KWIGR))
        ALPHA1 = -BETAW/(2.*BETAZ0*BETAZ0)
        ALPHAR2 = (1. - REF)/BETAZ0
        ALPHAI2 =-4.*NU*KWIGL**2*(1.-BETAW**2/2.)/(BETAZ0**2*KWIGR**2
        1 *GAM0)
C***    INITIALIZE PHASE AND AMPLITUDE
        TIME(1) = 1.
        GROE(1) = 0.
        TAU =1./FLOAT(NPART-1)
        TAUT = FLOAT(F)*TAU
        EWAV0 = 0.
        DO 60 J1=1,NMODE
        ARPRIM(J1) = 0.
```

```
        AIMPRIM(J1) = 0.
        AIMN(J1) = 0.
        ARN(J1) = EPS
        A(1,J1) = EPS
        EWAV0 = EWAV0 +(GAM0*KWIGR)**2*((KPOD(J1)-KWIGL)*A(1,J1)/
        1 KWIGL)**2/(4.*NU*(GAM0-1.))
        GROWTH(1,J1) = 0.
        WRITE(AB2(J1),105)JIFIX(OMEGA(J1)*BETAZ0/PI),NU,KPOD(J1)
        1 ,OMEGA(J1),NPLUS
        DO 100 J=1,NPART
        PSI0(J,J1) = PS00(J1,J-1) + (KPOD(J1) - OMEGA(J1))*(NPART - J)*T
AU
        UOLD(J,J1) = KPOD(J1) - OMEGA(J1)
100     CONTINUE
60      CONTINUE
        WRITE(6,102)
102     FORMAT(' THE WAVE AMPLITUDES ARE: ')
C***    BEGIN LOOP FOR TIME INCREMENTS
        DO 1000 N =2,NTIMES
        TIME(N) = 1. + FLOAT(N-1)*TAUT
        TRISE = (1. -EXP( (1.-TIME(N))/RISE ))
        DO 80 J1=1,NMODE
        DUM(J1) = 0.
        DUM1(J1) = 0.
80      CONTINUE
C***    BEGIN PART II: STEP AMPLITUDES AND PHASES
C***    COMPLETE SUM FOR AMPLITUDE STEP
        DO 200 J=1,NPART
        BETAZ = BETAZ0*(UOLD(J,1) + OMEGA(1))/KPOD(1)
        GAM(J)= SQRT( (1.-BETAZ*BETAZ)/(1./(GAM0*GAM0)+BETAW*BETAW/2.) )
        DUM2 = 0.
        DO 250 J1=1,NMODE
        DUM2 = DUM2 +(KPOD(1)*KPOD(J1)-OMEGA(J1)*BETAZ0*BETAZ0*
        1 (UOLD(J,1)+OMEGA(1)))*(ARN(J1)*COS(PSI0(J,J1))+
        1 AIMN(J1)*SIN(PSI0(J,J1)))+BETAZ0*BETAZ0*(UOLD(J,1)
        1 +OMEGA(1))*(ARPRIM(J1)*SIN(PSI0(J,J1))-AIMPRIM(J1)*
        1 COS(PSI0(J,J1)))
        DUM(J1) = DUM(J1) + COS(PSI0(J,J1))*TAU*GAM(J)
        DUM1(J1) = DUM1(J1) + SIN(PSI0(J,J1))*TAU*GAM(J)
250     CONTINUE
        PSIM(J,1) = PSI0(J,1) + TAUT*UOLD(J,1)
        UNEW(J,1) = UOLD(J,1) + TAUT*ALPHA1*GAM(J)*GAM(J)*DUM2
        DO 275 J1=2,NMODE
        PSIM(J,J1) = KPOD(J1)*PSIM(J,1)/KPOD(1)
        1 +(KPOD(J1)*OMEGA(1)/KPOD(1) -OMEGA(J1))*TIME(N)
        UNEW(J,J1) = KPOD(J1)*(UNEW(J,1)+OMEGA(1))/KPOD(1)-OMEGA(J1)
275     CONTINUE
200     CONTINUE
        EWAV(N) = 0.
        DO 280 J1=1,NMODE
        ARN1(J1)=ARN(J1)+TAUT*((ALPHAI2*AIMN(J1)+BETA1*DUM(J1))*TRISE/
        1 OMEGA(J1)-ALPHAR2*ARN(J1) )
        AIMN1(J1)=AIMN(J1)+TAUT*((-ALPHAI2*ARN(J1)+BETA1*DUM1(J1))*TRISE
        1 /OMEGA(J1)-ALPHAR2*AIMN(J1) )
        ARPRIM(J1) =(ALPHAI2*AIMN(J1)+BETA1*DUM(J1))*TRISE/OMEGA(J1)
        1 -ALPHAR2*ARN(J1)
        AIMPRIM(J1) =(-ALPHAI2*ARN(J1)+BETA1*DUM1(J1))*TRISE/OMEGA(J1)
        1 -ALPHAR2*AIMN(J1)
        A(N,J1) = SQRT(ARN1(J1)*ARN1(J1) + AIMN1(J1)*AIMN1(J1))
        GROWTH(N,J1) = 2.*(A(N,J1)-A(N-1,J1))/(TAUT*(A(N,J1)+A(N-1,J1)))
        EWAV(N) = EWAV(N)+(GAM0*KWIGR)**2*((KPOD(J1)-KWIGL)*A(N,J1)/
        1 KWIGL)**2/(4.*NU*(GAM0-1.))
280     CONTINUE
C***    END OF PART II A)
C***    BEGIN BOOKEEPING
        DO 400 J=1,NPART-F
```

```
          DO 375 J1=1,NMODE
          PSI0(J,J1) = PSIM(J+F,J1)
          UOLD(J,J1) = UNEW(J+F,J1)
375       CONTINUE
400       CONTINUE
          DO 500 J=F,1,-1
          DUM2 = 0.
          DO 475 J1=1,NMODE
          DUM2 = DUM2 +(KPOD(J1)*KPOD(1)-OMEGA(J1)*BETAZ0*BETAZ0
          1 *KPOD(1))*(ARN(J1)*COS(PS00(J1,NPART-J+(N-1)*F))+
          1 AIMN(J1)*SIN(PS00(J1,NPART-J+(N-1)*F))) +
          1 BETAZ0*BETAZ0*KPOD(1)*(ARPRIM(J1)*SIN(PS00(J1,NPART-J+(N-1)*F)
)
          1 -AIMPRIM(J1)*COS(PS00(J1,NPART-J+(N-1)*F)))
475       CONTINUE
          UOLD(NPART+1-J,1) =KPOD(1)-OMEGA(1)+(J-1)*TAU*ALPHA1*DUM2
          PSI0(NPART+1-J,1)=PS00(1,NPART-J+(N-1)*F)+(J-1)*TAU*(KPOD(1)-
          1 OMEGA(1) +UOLD(NPART+1-J,1) )/2.
          DO 495 J1=2,NMODE
          UOLD(NPART+1-J,J1) = KPOD(J1)*(UOLD(NPART+1-J,1)+OMEGA(1))/
          1 KPOD(1) - OMEGA(J1)
          PSI0(NPART+1-J,J1) =KPOD(J1)*PSI0(NPART+1-J,1)/KPOD(1)
          1 +(KPOD(J1)*OMEGA(1)/KPOD(1)-OMEGA(J1))*TIME(N)
495       CONTINUE
500       CONTINUE
C***      END OF PART II B) BOOKEEPING
          DO 600 J1=1,NMODE
          GROE(N) = (EWAV(N)-EWAV0)/(TAUT*EWAV0)
          ARN(J1) = ARN1(J1)
          AIMN(J1) = AIMN1(J1)
600       CONTINUE
          EWAV0 = EWAV(N)
C***      REPEAT PHASE AND AMPLITUDE INCREMENT
C***      FOR NEXT TIME STEP
1000      CONTINUE
          DO 2000 J1=1,NMODE
          YLAB = ' WAVE AMPLITUDE$'
          GLAB = ' WAVE AMPLITUDE VS. TIME MULTI$'
          CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
          CALL PWRIT(500,80,%REF(AB1),70,1,0,0)
          CALL PWRIT(500,48,%REF(AB2(J1)),72,1,0,0)
          CALL PWRIT(500,16, %REF(AB3),75,1,0,0)
          CALL EZXY(TIME,A(1,J1),NTIMES,%REF(AB4))
          YLAB = ' GROWTH RATE $'
          GLAB = ' GROWTH RATE NORMALIZED TO TRANSIT TIME$'
          CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
          CALL EZXY(TIME,GROWTH(1,J1),NTIMES,%REF(GLAB) )
2000      CONTINUE
          GLAB = 'FIELD ENERGY DENSITY VS. TIME$'
          YLAB = ' ENERGY DENSITY$'
          CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
          CALL EZXY(TIME,EWAV,NTIMES,%REF(GLAB))
          GLAB =' RATE OF CHANGE OF ENERGY$'
          YLAB = 'GROWTH RATE$'
          CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
          CALL EZXY(TIME,GROE,NTIMES,%REF(GLAB))
          STOP
          END
```

```
M   M   AAA   RRRR    AAA   BBBB   L      EEEEE
MM MM  A   A  R   R  A   A  B   B  L      E
M M M  A   A  R   R  A   A  B   B  L      E
M   M  A   A  RRRR   A   A  BBBB   L      EEEE
M   M  AAAAA  R R    AAAAA  B   B  L      E
M   M  A   A  R  R   A   A  B   B  L      E
M   M  A   A  R   R  A   A  BBBB   LLLLL  EEEEE


PPPP    AAA   RRRR   TTTTT  EEEEE  M   M  PPPP
P   P  A   A  R   R    T    E      MM MM  P   P
P   P  A   A  R   R    T    E      M M M  P   P
PPPP   A   A  RRRR     T    EEEE   M   M  PPPP
P      AAAAA  R R      T    E      M   M  P
P      A   A  R  R     T    E      M   M  P
P      A   A  R   R    T    EEEEE  M   M  P

           FFFFF   OOO   RRRR    ;;   4   4
           F      O   O  R   R   ;;   4   4
           F      O   O  R   R        4   4
           FFFF   O   O  RRRR    ;;   44444
           F      O   O  R R     ;;       4
        .. F      O   O  R  R     ;       4
        .. F       OOO   R   R   ;        4
```

File _VC$DRB1:[MARABLE.FEL]PARTEMP.FOR;4 (3928,1,0), last revised on 7-OCT-1984 15:18, is a 27 block sequential file owned by UIC [MARABLE]. The Records are variable length with a fixed control size of 2 bytes and implied (CR) carriage control. The longest record is 73 bytes.

Job PARTEMP (2004) queued to LN03_QUE on 21-MAR-1988 14:20 by user MARABLE, UIC [MARABLE], under account 4790 at priority 100, started on printer LTA8: on 21-MAR-1988 14:20 from queue VC_LN03B.

QQQQQQQQQQ 66666666666666666666666666666666666666666666666666 QQQQQQQQQQ
QQQQQQQQQQ Digital Equipment Corporation - VAX/VMS Version V4.6 QQQQQQQQQQ
QQQQQQQQQQ 66666666666666666666666666666666666666666666666666 QQQQQQQQQQ

```
C MAIN PROGRAM FOR COMPARISON OF DELTA FUNCTION AND STEP FUNCTION
C DISPERSIONS USING MULLER'S METHOD TO DETERMINE THE COMPLEX PART
C OF THE EIGENFREQUENCIES
        INTEGER N,NPREV,MAXIT,I,M,L,NPTS
        DOUBLE PRECISION EP1,EP2,DENS,WIG,LAM,BETA,PERP,BETA1,PERP1,GAM
        DOUBLE PRECISION DUM1,DUM2,DUM3,DUM4,X(9,50)
        DOUBLE PRECISION TS,TD,LINC,LINC2,LINC3,Y1(8),OMEG
        DOUBLE PRECISION ALP1,ALP2,DEL,GAMBAR,OMEG0
        CHARACTER*40 XLAB
        DOUBLE COMPLEX ZEROS(6),ROOT,PREVRT(20)
        EXTERNAL FN1,FN2
        LOGICAL FNREAL
        COMMON DENS,WIG,LAM,BETA,BETA1,PERP,PERP1,GAM,ALP1,ALP2
        OPEN (UNIT =1, NAME='OUTFILE' ,STATUS ='NEW')
        OPEN (UNIT =2, NAME='DATINI' ,STATUS ='NEW')
        FNREAL= .FALSE.
        MAXIT = 1000
        EP1 = 1.D-13
        EP2 = 1.D-13
        WRITE(6,1)
        WRITE(1,1)
1       FORMAT(' INPUT FOLLOWING DATA: DENS,WIG,LAM,LINC,LINC2,LINC3,BET

        1 ,PERP,DEL,NPTS ')
        READ(5,*)DENS,WIG,LAM,LINC,LINC2,LINC3,BETA,PERP,DEL,NPTS
        WRITE(1,*) DENS,WIG,LAM,LINC,LINC2,LINC3,BETA,PERP,DEL,NPTS
        WRITE(6,*) DENS,WIG,LAM,LINC,LINC2,LINC3,BETA,PERP,DEL,NPTS
        GAMBAR = DSQRT(1. + BETA*BETA + PERP + WIG)
        BETA = BETA/GAMBAR
        PERP = PERP/(GAMBAR*GAMBAR)
        WIG = WIG/(GAMBAR*GAMBAR)
        DENS = DENS/GAMBAR
        DEL = DEL/GAMBAR
        BETA1 = BETA/DSQRT(1.-PERP)
        PERP1 = PERP/(1.-PERP)
        GAM = 1./DSQRT(1.-PERP)
        TD = PERP*GAMBAR*(1. - .5*WIG)
        TS = .5*PERP1*GAMBAR*(1. - .5*WIG - PERP1/3.)/GAM
        WRITE(6,3) TD,TS
        WRITE(1,3) TD,TS
3       FORMAT(' DELTA TEMP = ',D12.3,' STEP TEMP = ',D12.3)
        WRITE(6,2)
        WRITE(1,2)
2       FORMAT(' LAM          2nd ROOT      DLD     DTP1D     DTM1D    DLS
        1   DTP1S   DTM1S')
        DO 1000 L = 1,NPTS
        IF( L .LE. 20) LAM = LAM + LINC
        IF( L .GT. 20 .AND. L .LE. 30) LAM = LAM + LINC2
        IF( L .GT. 30) LAM = LAM + LINC3
        ALP1 = DEL*LAM*(1.- BETA**2)
        ALP2 = DEL*LAM*GAM*((1.-BETA1**2)-PERP1*(1.-3.*BETA1**2)/4.)
        X(1,L) = LAM
        N = 6
        NPREV = 0
        ZEROS(1) = DCMPLX(0.,0.)
        IF (L .GT. 1) THEN
        ZEROS(1) = PREVRT(1)
        ZEROS(2) = PREVRT(2)
        ZEROS(3) = PREVRT(3)
        ZEROS(4) = PREVRT(4)
        ZEROS(5) = PREVRT(5)
        ZEROS(6) = PREVRT(6)
        END IF
        CALL MULLER(FN1,FNREAL,ZEROS,N,NPREV,MAXIT,EP1,EP2,1)
C       FIND EIGENVALUES WITH THE LARGEST GROWTH RATES
        CALL SORT1(N,ZEROS,DUM1,DUM2,DUM3,DUM4)
```

```
      Y1(1) = DUM3
      Y1(2) = DUM4
      X(2,L) = DUM1
      X(3,L) = DUM2
      PREVRT(1) = ZEROS(1)
      PREVRT(2) = ZEROS(2)
      PREVRT(3) = ZEROS(3)
      PREVRT(4) = ZEROS(4)
      PREVRT(5) = ZEROS(5)
      PREVRT(6) = ZEROS(6)
C  ****** X2 IS THE LARGEST GROWTH RATE IN THEN DATA SET FOR FULL DELTA
C ******  X3 IS THE REAL FREQUENCY CORESPONDING TO THE ABOVE
C *******  Y1 IS THE SECOND LARGEST DISTINCT IMAGINARY FREQUENCY
C ******* Y2 IS THE REAL FREQUENC CORRESPONDING TO THE ABOVE
      OMEG = DUM2 - BETA*LAM
      Y1(3) = LAM**2*(OMEG**2 - DENS*(1.-BETA**2))
      Y1(4) = DUM2**2 -(LAM +1.)**2 - DENS*(1.-PERP/2.)
      Y1(5) = DUM2**2 -(LAM-1.)**2 - DENS*(1. - PERP/2.)
      NPREV = 0
      ZEROS(1) = DCMPLX(0.,0.)
      IF (L .GT. 1) THEN
      ZEROS(1) = PREVRT(7)
      ZEROS(2) = PREVRT(8)
      ZEROS(3) = PREVRT(9)
      ZEROS(4) = PREVRT(10)
      ZEROS(5) = PREVRT(11)
      ZEROS(6) = PREVRT(12)
      END IF
      CALL MULLER(FN2,FNREAL,ZEROS,N,NPREV,MAXIT,EP1,EP2,1)
      CALL SORT1(N,ZEROS,DUM1,DUM2,DUM3,DUM4)
      X(4,L) = DUM1
      X(5,L) = DUM2
      PREVRT(7) = ZEROS(1)
      PREVRT(8) = ZEROS(2)
      PREVRT(9) = ZEROS(3)
      PREVRT(10) = ZEROS(4)
      PREVRT(11) = ZEROS(5)
      PREVRT(12) = ZEROS(6)
C ****  X4 IS THE LARGEST GROWTH RATE IN THE DATA SET FOR FULL STEP
C ****  X5 IS THE REAL FREQUENCY CORESPONDING TO THE ABOVE
      OMEG = DUM2 - BETA1*LAM*(1. - PERP1/4.)
      Y1(6) = DENS*GAM*((1.-BETA1**2)-PERP1*(1.-3.*BETA1**2)/4.)
      Y1(6) = LAM**2*(OMEG**2 - Y1(6))
      Y1(7) = DUM2**2-(LAM+1.)**2-DENS*(1.-PERP1/2.+(PERP1/2.)**2)*GAM
      Y1(8) = DUM2**2-(LAM-1.)**2-DENS*(1.-PERP1/2.+(PERP1/2.)**2)*GAM
      N = 4
      NPREV = 0
      ZEROS(1) = DCMPLX(0.,0.)
      IF (L .GT. 1) THEN
      ZEROS(1) = PREVRT(13)
      ZEROS(2) = PREVRT(14)
      ZEROS(3) = PREVRT(15)
      ZEROS(4) = PREVRT(16)
      END IF
      CALL MULLER(FN1,FNREAL,ZEROS,N,NPREV,MAXIT,EP1,EP2,2)
      CALL SORT1(N,ZEROS,DUM1,DUM2,DUM3,DUM4)
      X(6,L) = DUM1
      X(7,L) = BETA*LAM
      PREVRT(13) = ZEROS(1)
      PREVRT(14) = ZEROS(2)
      PREVRT(15) = ZEROS(3)
      PREVRT(16) = ZEROS(4)
C ***** X6 IS THE LARGEST GROWTH RATE FOR THE REFERENCE DELTA FUNCTION
C ****  X7 IS AN APPROXIMATION OF THE REAL FREQUENCY FOR SMALL DENSITIES
      NPREV = 0
      ZEROS(1) = DCMPLX(0.,0.)
```

```
          IF (L .GT. 1) THEN
          ZEROS(1) = PREVRT(17)
          ZEROS(2) = PREVRT(18)
          ZEROS(3) = PREVRT(19)
          ZEROS(4) = PREVRT(20)
          END IF
          CALL MULLER(FN2,FNREAL,ZEROS,N,NPREV,MAXIT,EP1,EP2,2)
          CALL SORT1(N,ZEROS,DUM1,DUM2,DUM3,DUM4)
          X(8,L) = DUM1
          X(9,L) = BETA1*LAM
          PREVRT(17) = ZEROS(1)
          PREVRT(18) = ZEROS(2)
          PREVRT(19) = ZEROS(3)
          PREVRT(20) = ZEROS(4)
C ****** X8 IS THE LARGEST GROWTH RATE FOR THE REFERENCE STEP FUNCTION
C ***** X9 IS AN APPROXIMATION OF THE REAL FREQUENCY FOR SMALL DENSITIES
          WRITE(1,5) LAM,(Y1(K),K=1,8)
          WRITE(6,5) LAM,(Y1(K),K=1,8)
1000      CONTINUE
          WRITE(1,6)
          WRITE(6,6)
6         FORMAT('    LAM        FULL DELTA        FULL STEP          REF DELTA
          1     REF STEP ')
          DO 2000 J = 1,NPTS
          WRITE(1,5)(X(K,J),K=1,9)
          WRITE(6,5)(X(K,J),K=1,9)
          WRITE(2,*)(X(K,J),K=1,9)
2000      CONTINUE
C ****    PLOT FULL DELTA vs. REF. DELTA
          XLAB = ' COMPARE FULL DELTA AND REF. DELTA'
          CALL QPICTR(X,18,NPTS,QY(3,11),QX(1),QMOVE(00),QXLAB(XLAB),QLABE
L(14))


C   ***   PLOT FULL DELTA vs. FULL STEP
          XLAB = ' COMPARE FULL DELTA AND FULL STEP'
          CALL QPICTR(X,18,NPTS,QY(3,7),QX(1),QMOVE(00),QXLAB(XLAB),QLABEL
 14))


C ****    PLOT REF. DELTA vs. REF STEP
          XLAB = ' COMPARE REF. DELTA AND REF. STEP'
          CALL QPICTR(X,18,NPTS,QY(11,15),QX(1),QMOVE(00),QXLAB(XLAB),QLAB
EL(14))


C ****    PLOT FULL STEP vs. REF STEP
          XLAB = ' COMPARE FULL STEP AND REF. STEP'
          CALL QPICTR(X,18,NPTS,QY(7,15),QX(1),QMOVE(00),QXLAB(XLAB),QLABE
L(14))

C ***** PLOT REAL FREQUENCIES FOR FULL DELTA AND STEP FUNCTION EQUILIBRI
A
          XLAB = ' COMPARE REAL FREQ. FOR DELTA & STEP'
          CALL QPICTR(X,18,NPTS,QY(5,9,13,17),QX(1),QMOVE(00),QXLAB(XLAB))



5         FORMAT(9D9.2)

          STOP
          END
          SUBROUTINE FN1(Z,FZ)
          DOUBLE COMPLEX Z,FZ(2),DLD,DTP1D,DTM1D,CHIBD,CHIAD,OMEG
          DOUBLE PRECISION DENS,WIG,LAM,BETA,BETA1,PERP,PERP1,GAM
          DOUBLE PRECISION ALP1,ALP2
```

```
      COMMON DENS,WIG,LAM,BETA,BETA1,PERP,PERP1,GAM,ALP1,ALP2
      OMEG = Z - BETA*LAM + ALP1*DCMPLX(0.,1.)
      DLD = LAM*LAM*(OMEG*OMEG-DENS*(1.-BETA*BETA))
      DTP1D = Z*Z -(LAM +1.)*(LAM +1.) -DENS*(1.-PERP/2.)
      DTM1D = Z*Z -(LAM -1.)*(LAM -1.) -DENS*(1.-PERP/2.)
      CHIBD = DENS*LAM*(BETA*OMEG*(1.-3.*PERP/2.)-LAM*(1.-BETA*BETA)
      1*(1.-PERP/2.))
      CHIAD = DENS*(2.*BETA*LAM*OMEG*(1.-PERP/2.)*(1.-3.*PERP/2.)-
      1 LAM*LAM*(1.-BETA*BETA)*(1.-PERP/2.)*(1.-PERP/2.) +
      1 (1.-3.*PERP)*OMEG*OMEG)
      FZ(1) =DLD*DTP1D*DTM1D + (WIG/2.)*(DTP1D +DTM1D)*LAM*LAM*(CHIAD
      1-DENS*DENS*((1.-3.*PERP)*(1.-BETA*BETA)+BETA*BETA*(1.-3.*PERP/2
.)
      1*(1.-3.*PERP/2.)))
      FZ(2)=DENS*(1.-BETA*BETA)*DLD*DTM1D-(WIG/2.)*CHIBD*CHIBD
      RETURN
      END
      SUBROUTINE FN2(Z,FZ)
      DOUBLE COMPLEX Z,FZ(2),DLS,DTP1S,DTM1S,CHIBS,CHIAS,OMEG
      DOUBLE PRECISION DENS,WIG,LAM,BETA,BETA1,PERP,PERP1,GAM
      DOUBLE PRECISION ALP1,ALP2
      COMMON DENS,WIG,LAM,BETA,BETA1,PERP,PERP1,GAM,ALP1,ALP2
      OMEG = Z - BETA1*LAM*(1.-PERP1/4.) + ALP2*DCMPLX(0.,1.)
      DLS=DENS*GAM*((1.-BETA1*BETA1)-PERP1*(1.-3.*BETA1*BETA1)/4.)
      DLS = LAM*LAM*(OMEG*OMEG - DLS)
      DTP1S =Z*Z-(LAM+1.)*(LAM+1.)-DENS*GAM*(1.-PERP1/2.+PERP1*PERP1/4
.)
      DTM1S =Z*Z-(LAM-1.)*(LAM-1.)-DENS*GAM*(1.-PERP1/2.+PERP1*PERP1/4
.)
      CHIBS =DENS*LAM*GAM*GAM*(BETA1*OMEG*(1.-3.*PERP1/2.)-LAM*
      1 (1.-PERP1/2.)*(1.-BETA1*BETA1-PERP1*(1.-3.*BETA1*BETA1)/4.))
      CHIAS=DENS*GAM*GAM*GAM*(2.*BETA1*LAM*OMEG*(1.-PERP1/2.)*(1.-3.*
      1 PERP1/2.)-LAM*LAM*((1.-PERP1/2.)*(1.-PERP1/2.)*(1.-BETA1*BETA1
-
      1 PERP1*(1.-3.*BETA1*BETA1)/4.))+(1.-9.*PERP1/4.)*OMEG*OMEG)
      FZ(1) = DLS*DTP1S*DTM1S+(WIG/2.)*(DTP1S+DTM1S)*LAM*LAM*(CHIAS -
      1 DENS*DENS*GAM**4*((1.-9.*PERP1/4.)*(1.-BETA1*BETA1-PERP1*(1.-3
.
      1*BETA1*BETA1)/4.)+BETA1*BETA1*(1.-3.*PERP1/2.)*(1.-3.*PERP1/2.)
))
      FZ(2)=DENS*GAM*(1.-BETA1*BETA1-PERP1*(1.-3.*BETA1*BETA1)/4.)*DLS
      1*DTM1S - (WIG/2.)*CHIBS*CHIBS
      RETURN
      END
      SUBROUTINE MULLER(FN,FNREAL,ZEROS,N,NPREV,MAXIT,EP1,EP2,M)
C DETERMINES UP TO N ZEROS OF THE FUNCTION SPECIFIED BY FN USING
C QUADRATIC INTERPOLATION, i.e. MULLER'S MEHTOD
      EXTERNAL FN1,FN2
      LOGICAL FNREAL
      INTEGER MAXIT,N,NPREV,KOUNT,L,M
      DOUBLE PRECISION EP1,EP2,EPS1,EPS2
      DOUBLE COMPLEX ZEROS(N),C,DEN,DIVDF1,DIVDF2,DVDF1P,FZR(2),FZRDFL
      1,FZRPRV,H,ZERO,SQR,HPREV,FN
C *********************  INPUT    ******************************
C   FN NAME OF SUBROUTINE, OF THE FORM FN(X,FX) WHICH FOR GIVEN X
C   RETURNS F(X), THIS MUST APPEAR IN AN EXTERNAL STATEMENT IN MAIN
C   CALLING PROGRAM.
C   FNREAL IS A LOGICAL VARIABLE, IF .TRUE. ALL APPROX. ARE TAKEN
C   TO BE REAL , ALLOWING THIS ROUTINE TO BE USED EVEN IF F(X) IS ONLY
C   DEFINED FOR REAL X.
C   ZEORS(1).... ZEROS(NPREV) CONTAINS PREVIOUSLY FOUND ZEROS OF THE
C   FUNCTION, PROVIDED NPREV .GT. 0
C    ZEROS(NPREV+1).... ZEROS(N) CONTAINS FIRS GUESS FOR THE ZEROS
C     TO BE FOUND
C    MAXIT IS THE MAXIMUM NUMBER OF FUNCTION EVALUATIONS ALLOWED @ ZERO.
C   EP1 ITERATION IS STOPPED IF ABS(H) .LT. EP1*ABS(ZR), WITH
```

```
C   H EQUAL TO THE LATEST CHANGE IN THE ZERO ESTIMATE
C   EP2 ALTHOUT THE EP1 CRITERION IS NOT MET, ITERATION IS STOPPED IF
C   ABS(F(ZERO)) .LT. EP2
C  N IS THE TOTAL NUMBER OF ZEROS TO BE FOUND
C   NPREV IS THE NUMBER OF ZEROS FOUND PREVIOUSLY
C********************    OUTPUT    ***************************
C   ZEROS(NPREV +1) ... ZEROS(N)  APPROXIMATIONS TO ZEROS
C           INITIALIZTION
        EPS1 = DMAX1(EP1,1.D-12)
        EPS2 = DMAX1(EP2,1.D-20)
c
        DO 500 I=NPREV +1,N
        KOUNT = 0
C    COMPUTE FIRST THREE ESTIMATES FOR ZERO AS ...
C     ZEROS(I)+.5,  ZEROS(I) - .5,    ZEROS(I)
401     ZERO = ZEROS(I)
        H = .5
        CALL DFLATE(FN,ZERO+.5,I,KOUNT,FZR,DVDF1P,ZEROS,L,M)
        IF(L .NE. 0) GO TO 401
        CALL DFLATE(FN,ZERO-.5,I,KOUNT,FZR,FZRPRV,ZEROS,L,M)
        IF(L .NE. 0) GO TO 401
        HPREV = -1.
        DVDF1P = (FZRPRV - DVDF1P)/HPREV
        CALL DFLATE(FN,ZERO,I,KOUNT,FZR,FZRDFL,ZEROS,L,M)
        IF(L .NE. 0) GO TO 401
C    DO WHILE KOUNT .LE. MAXIT OF H IS RELATIVELY BIG
C     OR FZR = F(ZERO) IS NOT SMALL
C      OR FZRDFL = FDFLATED(ZERO) IS NOT SMALL OR NOT MUCH
C     BIGGER THAN ITS PREVIOUS VALUE FZRPRV.
440     DIVDF1 = (FZRDFL - FZRPRV)/H
        DIVDF2 = (DIVDF1 - DVDF1P)/(H + HPREV)
        HPREV = H
        DVDF1P = DIVDF1
        C = DIVDF1 + H*DIVDF2
        SQR = C*C - 4.*FZRDFL*DIVDF2
        IF (FNREAL .AND. DREAL(SQR) .LT. 0.) SQR = (0.,0.)
        SQR = CDSQRT(SQR)
        IF (DREAL(C)*DREAL(SQR)+DIMAG(C)*DIMAG(SQR) .LT. 0.) THEN
          DEN = C - SQR
        ELSE
          DEN = C + SQR
        END IF
        IF(CDABS(DEN) .LE. 0.) DEN =(1.,0.)
        H = -2.*FZRDFL/DEN
        FZRPRV = FZRDFL
        ZERO = ZERO + H
        IF(KOUNT .GT. MAXIT) GO TO 499
470     CALL DFLATE(FN,ZERO,I,KOUNT,FZR,FZRDFL,ZEROS,L,M)
        IF(L .NE. 0) GO TO 401
C       CHECK FOR CONVERGENCE
        IF(CDABS(H) .LT. EPS1*CDABS(ZERO)) GO TO 499
        IF(DMAX1(CDABS(FZR(M)),CDABS(FZRDFL)) .LT. EPS2) GO TO 499
C       CHECK FOR DIVERGENCE
        IF(CDABS(FZRDFL) .GE. 10.*CDABS(FZRPRV)) THEN
        H = H/2.
        ZERO = ZERO - H
        GO TO 470
        ELSE
        GO TO 440
        END IF
499     ZEROS(I) = ZERO
500     CONTINUE
        RETURN
        END
        SUBROUTINE DFLATE(FN,ZERO,I,KOUNT,FZERO,FZRDFL,ZEROS,L,M)
C       TO BE CALLED BY MULLER
```

```
      INTEGER I,KOUNT,J,L,M
      DOUBLE COMPLEX FZERO(2),FZRDFL,ZERO,ZEROS(8),DEN
      L= 0
      KOUNT = KOUNT + 1
      CALL FN(ZERO,FZERO)
      FZRDFL = FZERO(M)
      IF(I .LT. 2) RETURN
      DO 410 J=2,I
      DEN = ZERO - ZEROS(J-1)
      IF(CDABS(DEN) .EQ. 0.) THEN
      ZEROS(I) = ZERO * 1.001
      L= 1
      RETURN
      ELSE
      FZRDFL = FZRDFL/DEN
      END IF
410   CONTINUE
      RETURN
      END
      SUBROUTINE SORT1(N,ZEROS,DUM1,DUM2,DUM3,DUM4)
      DOUBLE COMPLEX ZEROS(N)
      DOUBLE PRECISION DUM1,DUM2,DUM3,DUM4
      INTEGER I,J,K,N
C     FIND THE LARGEST GROWTH RATE IN THE DATA SET ZEROS(N)
      DUM1 =DIMAG(ZEROS(1))
      J = 1
      DO 100 I = 2,N
      IF (DUM1 .GT. DIMAG(ZEROS(I))) THEN
      DUM1 = DUM1
      J = J
      ELSE
      DUM1 = DIMAG(ZEROS(I))
      J = I
      END IF
100   CONTINUE
      DUM2 = DREAL(ZEROS(J))
      IF(DUM1 .LE. 0.) DUM1 = 0.
C     DUM1  IS THE LARGEST GROWTH RATE IN THE DATA SET
C     DUM2 IS THE REAL FREQUENCY CORRESPONDING TO THE MAX GROWTH
C   NEXT FIND THE NEXT LARGEST GROWTH RATE THAT IS NOT EQUAL IN
C   MAGNITUDE TO THE FIRST GROWTH RATE
      IF(J .EQ. 1) J = 3
      M = J - 1
      DUM3 = DIMAG(ZEROS(M))
      DO 300 I=1,N
      IF(DIMAG(ZEROS(I)).GE.DUM3.AND.DIMAG(ZEROS(I)).NE.
      1 DUM1) THEN
      DUM3 = DIMAG(ZEROS(I))
      K = I
      END IF
300   CONTINUE
      DUM4 = DREAL(ZEROS(K))
C     DUM3 IS THE SECOND LARGEST DISTINCT GROWTH RATE
C     DUM4 IS THE REAL FREQUENCY CORRESPONDING TO THE ABOVE
      RETURN
      END
```

```
0000000000     44444444444444444444444444444444444444444444444444     0000000000
0000000000     Digital Equipment Corporation - VAX/VMS Version V4.6     0000000000
0000000000     44444444444444444444444444444444444444444444444444     0000000000
```

```
M    M    AAA    RRRR     AAA    BBBB    L        EEEEE
MM  MM  A    A   R    R  A    A  B    B  L        E
M M M   A    A   R    R  A    A  B    B  L        E
M    M  A    A   RRRR    A    A  BBBB    L        EEEE
M    M  AAAAA    R R     AAAAA   B    B  L        E
M    M  A    A   R   R   A    A  B    B  L        E
M    M  A    A   R    R  A    A  BBBB    LLLLL    EEEEE

RRRR    EEEEE    SSSS   TTTTT    AAA     RRRR    TTTTT
R    R  E        S         T     A   A   R    R    T
R    R  E        S         T     A   A   R    R    T
RRRR    EEEE     SSS       T     A   A   RRRR      T
R R     E            S     T     AAAAA   R R       T
R   R   E            S     T     A   A   R   R     T
R    R  EEEEE   SSSS       T     A   A   R    R    T

                 CCCC   FFFFF   TTTTT     ;;     1
                C       F         T       ;;    11
                C       F         T              1
                C       FFFF      T       ;;     1
                C       F         T       ;;     1
           ..   C       F         T        ;     1
           ..    CCCC   F         T       ;     111
```

ile _VC$DRB1:[MARABLE]RESTART.CFT;1 (4038,4,0), last revised on 21-MAR-1988 14:
6, is a 21 block sequential file owned by UIC [MARABLE]. The records are varia
ble length with implied (CR) carriage control. The longest record is 72 bytes.

ob RESTART (2000) queued to LN03_QUE on 21-MAR-1988 14:09 by user MARABLE, UIC
[MARABLE], under account 4790 at priority 100, started on printer LTA7: on
21-MAR-1988 14:09 from queue VC_LN03A.

```
0000000000     44444444444444444444444444444444444444444444444444     0000000000
0000000000     Digital Equipment Corporation - VAX/VMS Version V4.6     0000000000
0000000000     44444444444444444444444444444444444444444444444444     0000000000
```

```
C***      RESTART CODE
 ***   CODE TO EVALUATE TEMPORAL EVOLUTION OF THE SPRECTRA
 ***   OF UNSTABLE MODES IN A HELICAL WIGGLER FREE ELECTRON LASER
C***   DELETION OF FIRST TRANSIT TIME
C***   FIELD AND PARTICLE EQUATIONS ARE EVOLVED BY ADAMS-BASHFORTH
 ***   METHOD WITH INITALIZATION BY RUNGE-KUTTA METHOD
C***   REFORMULATION OF THE PARTICLE PHASE  3/13
C***      CONVERSION TO CRAY FORTRAN
 ***   INCLUSION OF FREQUENCY SHIFT ERROR CHECK IN ADAMS-BASHFORTH
 ***   EQUATION SOLVER
C***   REPLACE EXPRESSION FOR THE DERIVATIVES WITH THE FUNCTIONAL
C***   EVALUATION OF THE DIFFERENTIAL EQUATION
 ***   PLOT DATA AFTER EVERY 10 CALCULATIONS
C***   OUTPUT DATA IF TIME LIMIT IS APPROACHED
C***   MODIFICATIONS TO PRODUCE RESTART DATA  8/1
       REAL BETAZ0,BETA0,KPOD(20),OMEGA(20),BETAZ,GAM
       REAL CTHET(20),PSI(20,3000,5),U0(3000,5),TIM
       REAL TEMP(3000),TEMP1(3000),ATEMP(20),ATEMP1(20),THETA(20,5)
       REAL TP(20,5),TIME(5000),TPLOT(6)
       REAL FREQ(5000,20),EWAV(5000),A(20,5),APP(20,5)
       REAL PHI1(20,5),PHI2(20,5),KWIGL,PSII(600,6),U0I(600,6)
     1 ,CORA(20),TPC(20),PLAI(5000,20)
       REAL KWIGR,NU,NUR,NUI,FILL,PLAR(5000,20)
       REAL F11(20),F12(20),F13(20),F14(20),F15(20),F21(20),F22(20)
       REAL F23(20),F24(20),F25(20)
       INTEGER F,MM,JP,J,J1,K,MAXIT,TLIM,NCOUNT,NLAST
       COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
       COMMON/BLK3/TIM,NPART,N,RISE,TAU,BETA1,BETA2,F,NUI,NUR
       COMMON/BLK4/ ALPHA1,ALPHA2,NMODE
       COMMON/BLK5/PSI,U0,PHI1,PHI2,A,THETA,APP,TP
       PARAMETER (PI=3.1415926535)
       PS00(J1,J) = -OMEGA(J1)*FLOAT(J-1)*TAU
       TRISE(N) = 1. -RISE*(EXP((-TIM+1.)/RISE) -1.)
       OPEN(UNIT=1,FILE='FILE1',FORM='UNFORMATTED',STATUS='OLD')
       OPEN(UNIT=2,FILE='FILE2',FORM='UNFORMATTED',STATUS='OLD')
       OPEN(UNIT=3,FILE='FILE3',FORM='UNFORMATTED',STATUS='OLD')
       OPEN(UNIT=4,FILE='FILE4',FORM='UNFORMATTED',STATUS='NEW')
       OPEN(UNIT=5,FILE='FILE5',FORM='UNFORMATTED',STATUS='NEW')
       OPEN(UNIT=7,FILE='FILE7',FORM='UNFORMATTED',STATUS='NEW')
       READ(1) TIME,FREQ,EWAV,PLAR,PLAI,KPOD,OMEGA,KWIGR,NU,
     1 GAM0,BETAW,RISE,FILL,REF,EPS,PHASE,ERROR,BETAZ0,ERROR2,
     1 PSII,U0I,TPLOT,TLIM,BETA0
       READ(2) NWIG,NPART,F,NMODE,NPLUS,MAXIT,NSEP,NTIMES,NCOUNT,
     1 NLAST
       READ(3) PSI,U0,F11,F12,F13,F21,F22,F23,A,APP,THETA,
     1 TP,PHI1,PHI2
       NCOUNT = NCOUNT + 1
       FORMAT( ' THE PRED.-CORR. METHD FAILED TO CONVERGE ON STEP',2X,
     1   I7,' AFTER ',I4,2X,' INTERATIONS',2X,'ON MODE ',I3)
       KWIGL = 2.*FLOAT(NWIG)*PI
       BETA1 = 2.*FILL*NU*KWIGL**2*BETA0*BETAW/(KWIGR**2*BETAZ0**3)
       NUR = (1.-REF)/BETAZ0
       NUI = -4.*FILL*NU*KWIGL**2*(1.-BETAW**2/2.)/(GAM0*BETAZ0*
     1   KWIGR**2*BETAZ0*OMEGA(1))
       BETA2 = 8.*NU*BETA0*KWIGL**2/(BETAZ0*KWIGR**2)
       BETA2 = 0.
       ALPHA1 = KPOD(1)/BETAZ0
       ALPHA2 = .5*BETAW*GAM0*KPOD(1)/BETAZ0**2
       TAU = 1./FLOAT(NPART -1)
       TAUT = FLOAT(F)*TAU
C***   NOW EVOLVE PARTICLES AND FIELDS WITH ADAMS-BASHFORTH PREDICTOR
 ***   CORRECTOR METHOD USING THE RESULTS OF THE FOUR PREVIOUS TIMES
C***   AS INITIAL CONDITIONS
       DO 7000 N=NLAST+1,NTIMES
       TIM = 1. + FLOAT(N-1)*TAUT
       DO 5100 J1=1,NMODE
```

```
5100   CALL EVOLV(J1,F24,F14,2)
       DO 5200 JP=1,NPART-4*F
       CALL F4(JP+F,FOUT,2)
       CALL F4(JP+2*F,FOUT1,3)
       CALL F4(JP+3*F,FOUT2,4)
       CALL F4(JP+4*F,FOUT3,5)
       U0(JP,1) = U0(JP+F,2) + TAUT*(55.*FOUT -59.*FOUT1 +37.*FOUT2
     1   -9.*FOUT3)/24.
       PSI(1,JP,1) = PSI(1,JP+F,2) + TAUT*(55.*U0(JP+F,2) -
     1   59.*U0(JP+2*F,3)+37.*U0(JP+3*F,4)-9.*U0(JP+4*F,5))/24.
       TEMP(JP) = 19.*FOUT -5.*FOUT1 + FOUT2
5200   TEMP1(JP) = 19.*U0(JP+F,2)-5.*U0(JP+2*F,3)+U0(JP+3*F,4)
       DO 5300 JP=NPART-4*F+1,NPART-F
       CALL F4(JP+F,FOUT,2)
       U0(JP,1) = U0(JP+F,2) + TAUT*FOUT
5300   PSI(1,JP,1) = PSI(1,JP+F,2)+TAUT*(U0(JP+F,2) +U0(JP,1))/2.
       DO 5400 JP = NPART-F+1,NPART
       J = JP + N1*F
       U0(JP,1) = KPOD(1) - OMEGA(1)
5400   PSI(1,JP,1) = PS00(1,J)+ FLOAT(NPART-JP)*TAU*U0(JP,1)
       DO 5500 J1=1,NMODE
       A(J1,1) = A(J1,2)+TAUT*(55.*F14(J1)-59.*F13(J1)+37.*F12(J1)
     1   - 9.*F11(J1))/24.
       THETA(J1,1) = THETA(J1,2) + TAUT*(55.*F24(J1)-59.*F23(J1)
     1   +37.*F22(J1) - 9.*F21(J1))/24.
       APP(J1,1) = F14(J1)
C***   THETA PRIME IS THE AVERAGE OF THE DISCRETE AND FUNCTIONAL
C***   VALUES OF THE DERIVATIVE
       TP(J1,1) = F24(J1)
       ATEMP(J1) = 19.*F14(J1) - 5.*F13(J1) + F12(J1)
       ATEMP1(J1) = 19.*F24(J1) - 5.*F23(J1) + F22(J1)
5500   CONTINUE
       DO 5800 M=1,MAXIT
       DO 5700 J1=1,NMODE
       CALL EVOLV(J1,F25,F15,1)
       CORA(J1) = A(J1,1)
       A(J1,1) = A(J1,2) + TAUT*(9.*F15(J1) + ATEMP(J1))/24.
       CTHET(J1) = THETA(J1,1)
       THETA(J1,1) = THETA(J1,2) + TAUT*(9.*F25(J1) + ATEMP1(J1))/24.
       TPC(J1) = F25(J1)
       TP(J1,1) = F25(J1)
5700   APP(J1,1) = F15(J1)
       IF( MOD(N,10) .EQ. 0 )THEN
       NPLOT = INT(N/10) + 1
       TIME(NPLOT) = TIM
       EWAV(NPLOT) = 0.
        END IF
       DO 5600 JP =1,NPART-4*F
       CALL F4(JP,FOUT,1)
       U0(JP,1) = U0(JP+F,2) +TAUT*(9.*FOUT +TEMP(JP))/24.
5600   PSI(1,JP,1) = PSI(1,JP+F,2) +TAUT*(9.*U0(JP,1)+TEMP1(JP))/24.
       DO 5750 J1=1,NMODE
       IF( ABS( A(J1,1)-CORA(J1))/ABS(CORA(J1)) .GT. ERROR .OR.
     1  ABS(THETA(J1,1)-CTHET(J1))/ABS(CTHET(J1)) .GT. ERROR .OR.
     1  ABS(TP(J1,1)-TPC(J1))/ABS(TP(J1,1)) .GT. ERROR2)THEN
       GO TO 5799
       ELSE
       IF( MOD(N,10) .EQ. 0)THEN
       NPLOT = INT(N/10) + 1
       PLAR(NPLOT,J1) = A(J1,1)*SIN(THETA(J1,1))
       PLAI(NPLOT,J1) = -A(J1,1)*COS(THETA(J1,1))
       FREQ(NPLOT,J1) = TP(J1,1)
       EWAV(NPLOT) = EWAV(NPLOT) + KWIGR**2*((BETAZ0*OMEGA(J1)*
     1   A(J1,1))**2 + KPOD(J1)**2*(PHI1(J1,1)**2+PHI2(J1,1)**2)
     1   /4.)/(4.*NU*(GAM0-1.)*KWIGL**2 )
       END IF
```

```
      END IF
 750  CONTINUE
      GO TO 5850
5799  IF(M .EQ. MAXIT)WRITE(6,1)N,M
5800  CONTINUE
 850  DO 6200 K=4,1,-1
      DO 5900 J1=1,NMODE
      A(J1,K+1) = A(J1,K)
      APP(J1,K+1) = APP(J1,K)
      THETA(J1,K+1) = THETA(J1,K)
5900  TP(J1,K+1) = TP(J1,K)
      DO 6100 JP=1,NPART-F
      U0(JP+F,K+1) = U0(JP,K)
      PSI(1,JP+F,K+1) = PSI(1,JP,K)
6100  CONTINUE
 200  CONTINUE
      DO 6300 J1=1,NMODE
      F11(J1) = F12(J1)
      F12(J1) = F13(J1)
      F13(J1) = F14(J1)
      F21(J1) = F22(J1)
      F22(J1) = F23(J1)
 300  F23(J1) = F24(J1)
      CALL SECOND(CPU)
      IF( CPU .GE. .95*FLOAT(TLIM) ) GO TO 7001
7000  CONTINUE
 001  NLAST = N
      JJ=0
      DO 7002 JP=5,NPART,5
      JJ = JJ + 1
      U0I(JJ,NCOUNT+1) = U0(JP,2)
 002  PSII(JJ,NCOUNT+1) = PSI(1,JP,2)
      TPLOT(NCOUNT+1) = TIM
      WRITE(4) TIME,FREQ,EWAV,PLAR,PLAI,KPOD,OMEGA,KWIGR,
     1 NU,GAM0,BETAW,RISE,FILL,REF,EPS,PHASE,
     1 ERROR,BETAZ0,ERROR2,PSII,U0I,TPLOT,TLIM,BETA0
      WRITE(5)NWIG,NPART,F,NMODE,NPLUS,MAXIT,NSEP,NTIMES
     1 ,NCOUNT,NLAST
      WRITE(7) PSI,U0,F11,F12,F13,F21,F22,F23,A,APP,THETA,
     1 TP,PHI1,PHI2
      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
      CLOSE(UNIT=3)
      CLOSE(UNIT=4)
      CLOSE(UNIT=5)
      CLOSE(UNIT=7)
      END
      SUBROUTINE F4(JP,FOUT,MM)
      REAL BETAZ,BETAZ0,GAM,KPOD(20),OMEGA(20)
      REAL PSI(20,3000,5),U0(3000,5),PHI1(20,5),PHI2(20,5),A(20,5),
     1   APP(20,5),TP(20,5),THETA(20,5),FOUT
      INTEGER JP,MM,NMODE
      COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
      COMMON/BLK5/PSI,U0,PHI1,PHI2,A,THETA,APP,TP
      COMMON/BLK4/ALPHA1,ALPHA2,NMODE
      BETAZ = BETAZ0*(U0(JP,MM) + OMEGA(1) )/KPOD(1)
      GAM = SQRT( (1.+(GAM0*BETAW)**2)/(1.-BETAZ*BETAZ))
      SUM1 =KPOD(1)*(PHI2(1,MM)*COS(PSI(1,JP,MM)-THETA(1,MM))
     1   -PHI1(1,MM)*SIN(PSI(1,JP,MM)-THETA(1,MM)))
      SUM2 =(KPOD(1)-BETAZ0*BETAZ*(OMEGA(1)+TP(1,MM)))*A(1,MM)*
     1   SIN(PSI(1,JP,MM)-THETA(1,MM))-BETAZ0*BETAZ*APP(1,MM)*
     1   COS(PSI(1,JP,MM)-THETA(1,MM))
      DO 100 J1=2,NMODE
      SUM1=SUM1 +KPOD(J1)*(PHI2(J1,MM)*COS(PSI(J1,JP,MM)-THETA(J1,MM))
     1   -PHI1(J1,MM)*SIN(PSI(J1,JP,MM)-THETA(J1,MM)))
      SUM2=SUM2+(KPOD(J1)-BETAZ0*BETAZ*(OMEGA(J1)+TP(J1,MM)))*A(J1,MM)
```

```
     1     *SIN(PSI(J1,JP,MM)-THETA(J1,MM))-BETAZ0*BETAZ*APP(J1,MM)*
     1      COS(PSI(J1,JP,MM)-THETA(J1,MM))
 100      CONTINUE
          FOUT = ALPHA1*SUM1*(1.-BETAZ*BETAZ)/GAM + ALPHA2*SUM2/(GAM*
     1      GAM)
          RETURN
          END
          SUBROUTINE EVOLV(J1,K2,K1,MM)
          REAL BETAZ0,BETAZ,GAM,KPOD(20),OMEGA(20)
          REAL A(20,5),PHI1(20,5),PHI2(20,5),PSI(20,3000,5),U0(3000,5)
          REAL K2(20),K1(20),TIM,THETA(20,5),APP(20,5),TP(20,5)
          REAL NUI,NUR
          INTEGER J1,MM,N,F,NPART
          COMMON/BLK1/BETAZ0,GAM0,BETAW,KPOD,OMEGA
          COMMON/BLK5/PSI,U0,PHI1,PHI2,A,THETA,APP,TP
          COMMON/BLK3/TIM,NPART,N,RISE,TAU,BETA1,BETA2,F,NUI,NUR
          DUM1 = 0.
          DUM2 = 0.
          DUM3 = 0.
          DUM4 = 0.
          DO 100 JP =1,NPART
          J = JP + (N-1)*F
          TRISE = 1. - EXP(-FLOAT(J-1)*TAU/RISE)
          BETAZ = BETAZ0*(U0(JP,MM) + OMEGA(1) )/KPOD(1)
          GAM = SQRT( (1.+(GAM0*BETAW)**2)/(1.-BETAZ*BETAZ) )
          PSI(J1,JP,MM)=KPOD(J1)*(PSI(1,JP,MM)+OMEGA(1)*TIM)/KPOD(1)
     1      - OMEGA(J1)*TIM
          DUM1 = DUM1 + COS(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE*GAM0/GAM
          DUM2 = DUM2 + SIN(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE*GAM0/GAM
          DUM3 = DUM3 + COS(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE
 100      DUM4 = DUM4 + SIN(PSI(J1,JP,MM)-THETA(J1,MM))*TAU*TRISE
          K1(J1) = -NUR*A(J1,MM)/2.- BETA1*DUM2/OMEGA(J1)
          K2(J1) = -NUI/2. + BETA1*DUM1/(A(J1,MM)*OMEGA(J1))
          PHI1(J1,MM) = - BETA2*DUM3/KPOD(J1)**2
          PHI2(J1,MM) = - BETA2*DUM4/KPOD(J1)**2
          RETURN
          END
```

```
JJJJJJJJJJ    9999999999999999999999999999999999999999999999999999    JJJJJJJJJJ
JJJJJJJJJJ    Digital Equipment Corporation - VAX/VMS Version V4.6    JJJJJJJJJJ
JJJJJJJJJJ    9999999999999999999999999999999999999999999999999999    JJJJJJJJJJ


               M   M    AAA    RRRR    AAA    BBBB   L      EEEEE
               MM MM   A   A   R   R  A   A   B   B  L      E
               M M M   A   A   R   R  A   A   B   B  L      E
               M   M   A   A   RRRR   A   A   BBBB   L      EEEE
               M   M   AAAAA   R R    AAAAA   B   B  L      E
               M   M   A   A   R  R   A   A   B   B  L      E
               M   M   A   A   R   R  A   A   BBBB   LLLLL  EEEEE


          SSSSSSSS   PPPPPPPP    RRRRRRRR      AAAAAA    DDDDDDDD    BBBBBBBB
          SSSSSSSS   PPPPPPPP    RRRRRRRR      AAAAAA    DDDDDDDD    BBBBBBBB
        SS           PP      PP  RR      RR  AA      AA  DD      DD  BB      BB
        SS           PP      PP  RR      RR  AA      AA  DD      DD  BB      BB
        SS           PP      PP  RR      RR  AA      AA  DD      DD  BB      BB
        SS           PP      PP  RR      RR  AA      AA  DD      DD  BB      BB
          SSSSSS     PPPPPPPP    RRRRRRRR    AA      AA  DD      DD  BBBBBBBB
          SSSSSS     PPPPPPPP    RRRRRRRR    AA      AA  DD      DD  BBBBBBBB
                SS   PP          RR  RR      AAAAAAAAAA  DD      DD  BB      BB
                SS   PP          RR  RR      AAAAAAAAAA  DD      DD  BB      BB
                SS   PP          RR    RR    AA      AA  DD      DD  BB      BB
                SS   PP          RR    RR    AA      AA  DD      DD  BB      BB
        SSSSSSSS     PP          RR      RR  AA      AA  DDDDDDDD    BBBBBBBB
        SSSSSSSS     PP          RR      RR  AA      AA  DDDDDDDD    BBBBBBBB


                     FFFFFFFFFF    OOOOOO    RRRRRRRR        ;;;;        11
                     FFFFFFFFFF    OOOOOO    RRRRRRRR        ;;;;        11
                     FF          OO      OO  RR      RR      ;;;;      1111
                     FF          OO      OO  RR      RR      ;;;;      1111
                     FF          OO      OO  RR      RR                  11
                     FF          OO      OO  RR      RR                  11
                     FFFFFFFF    OO      OO  RRRRRRRR        ;;;;        11
                     FFFFFFFF    OO      OO  RRRRRRRR        ;;;;        11
                     FF          OO      OO  RR  RR          ;;;;        11
                     FF          OO      OO  RR  RR          ;;;;        11
             ....    FF          OO      OO  RR    RR          ;;        11
             ....    FF          OO      OO  RR    RR          ;;        11
             ....    FF            OOOOOO    RR      RR      ;;        111111
             ....    FF            OOOOOO    RR      RR      ;;        111111


File _VC$DRB1:[MARABLE.TWSTAG]SPRADB.FOR;1 (4143,1,0), last revised on
14-DEC-1984 10:33, is a 10 block sequential file owned by UIC [MARABLE].  The
records are variable length with a fixed control size of 2 bytes and implied
(CR) carriage control.  The longest record is 71 bytes.

Job SPRADB (1985) queued to LN03_QUE on 21-MAR-1988 13:48 by user MARABLE, UIC
[MARABLE], under account 4790 at priority 100, started on printer LTA7: on
21-MAR-1988 13:48 from queue VC_LN03A.

JJJJJJJJJJ    9999999999999999999999999999999999999999999999999999    JJJJJJJJJJ
JJJJJJJJJJ    Digital Equipment Corporation - VAX/VMS Version V4.6    JJJJJJJJJJ
JJJJJJJJJJ    9999999999999999999999999999999999999999999999999999    JJJJJJJJJJ
```

```
C***    TEST PROGRAM FOR MODEL EQUATIONS IN HIGH POWER
C***    FEL PROBLEM.
C***    REVISION 12/6  NEW FORMULATION OF WAVE EQUATION
C***    REVISION 12/14 ADDITION OF DOUBLE PRECISION
        REAL A(1200),EPS,TAU,OMEGA
        REAL TIME(1200),KPOD,GAM(400)
        REAL GROWTH(1200),KWIGR,KWIGL,NU,FEQ(1200),RISE
        DOUBLE PRECISION PSIM(400),PSI0(400),BETA1,DUM,DUM1,PS00
        DOUBLE PRECISION UOLD(400),UNEW(400),FEQD,AD,GROW,THETAP
        DOUBLE PRECISION ALPHA1,AMAG,THETA
        INTEGER J,NPART,NTIMES,N,F
        CHARACTER*40 GLAB,XLAB,YLAB
        CHARACTER*80 AB1,AB2,AB3
        PS00(J) = -OMEGA*DFLOAT(J)*TAU
        OPEN(UNIT=2,NAME='OUT.DAT',STATUS='NEW')
        GLAB= 'WAVE AMPLITUDE VS. TIME$'
        XLAB =' TIME$'
        YLAB = ' WAVE AMPLITUDE$'
        WRITE(6,101)
101     FORMAT(' INPUT NO. OF PART.,NO. OF INTERAT.,GAM,BETAWIG,
        1 KWIGr,BUDKER,KPOD,OMEGA,KWIGL,EPS,F,RISE ')
        READ(5,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,KPOD,OMEGA,KWIGL,EPS,F
RISE
        WRITE(6,103)
        WRITE(2,103)
103     FORMAT(' THE INPUT DATA:NPART,NTIMES,GAM,BETAW,KWIGR,NU,KPOD
        1 ,OMEGA,KWIGL,EPS,F,RISE IS: ')
        WRITE(6,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,KPOD,OMEGA,KWIGL,EPS,
F,RISE
        WRITE(2,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,KPOD,OMEGA,KWIGL,EPS,
,RISE
104     FORMAT('NPART=',I4,3X,'NTIMES=',I4,3X,'GAM=',F8.4,3X,
        1 'BETAWIG=',F8.4)
105     FORMAT('KWIGR=',F8.4,3X,'NU=',E10.4,3X,'KPOD=',E10.4
        1,3X,'OMEGA=',E10.4)
106     FORMAT('KWIGL=',F8.4,3X,'EPS=',E10.4,3X,'F=',I3,3X,
        1'RISE=',F8.4)
        WRITE(AB1,104)NPART,NTIMES,GAM0,BETAW
        WRITE(AB2,105)KWIGR,NU,KPOD,OMEGA
        WRITE(AB3,106)KWIGL,EPS,F,RISE
        BETA0 = SQRT(1. -1./(GAM0*GAM0))
        BETAZ0 = SQRT(BETA0*BETA0 - BETAW*BETAW/2.)
        BETA1 = 4.*NU*BETAW*(KWIGL)**2/(BETAZ0*BETA0*GAM0*OMEGA
        1 *(KWIGR*KWIGR))
        ALPHA1 = -BETAW/(2.*BETAZ0*BETAZ0)
C***    INITIALIZE PHASE AND AMPLITUDE
        AMAG = EPS
        THETA = 0.
        A(1) = EPS
        TIME(1) = 1.
        TAU =1./FLOAT(NPART-1)
        DO 100 J=1,NPART
        PSI0(J) = PS00(J-1) + (KPOD - OMEGA)*(NPART - J)*TAU
        UOLD(J) = KPOD - OMEGA
100     CONTINUE
        WRITE(6,102)
        WRITE(2,102)
102     FORMAT(' THE WAVE AMPLITUDES ARE: ')
C***    BEGIN LOOP FOR TIME INCREMENTS
        DO 1000 N =2,NTIMES
        TIME(N) = 1. + FLOAT(N-1)*FLOAT(F)/(NPART -1)
C***    BEGIN PART II: STEP AMPLITUDES AND PHASES
C***    COMPLETE SUM FOR AMPLITUDE STEP
        DUM = 0.
        DUM1 = 0.
        DO 200 J=1,NPART
```

```
          BETAZ = BETAZ0*(UOLD(J) + OMEGA)/KPOD
          GAM(J)= SQRT( (1.-BETAZ*BETAZ)/(1./(GAM0*GAM0)+BETAW*BETAW/2.) )
          DUM = DUM + DCOS(PSI0(J) - THETA)*TAU*GAM(J)
          DUM1 = DUM1 + DSIN(PSI0(J) - THETA)*TAU*GAM(J)
200       CONTINUE
          GROW = BETA1*(1.-EXP((1.-TIME(N))/RISE))*DUM/AMAG
          THETAP = BETA1*(1.-EXP((1.-TIME(N))/RISE))*DUM1/AMAG
          AD = AMAG + DFLOAT(F)*TAU*GROW*AMAG
          A(N) = AD
          DO 300 J=1,NPART
          PSIM(J) = PSI0(J) + DFLOAT(F)*TAU*UOLD(J)
          UNEW(J) = UOLD(J) + DFLOAT(F)*TAU*ALPHA1*GAM(J)*GAM(J)*(
          1 (KPOD*KPOD-BETAZ0*BETAZ0*OMEGA*(UOLD(J)+OMEGA))*
          1 (AMAG*DCOS(PSI0(J) - THETA)) +
          1 BETAZ0*BETAZ0*(UOLD(J)+OMEGA)*(AMAG*GROW*DSIN(PSI0(J)-THETA)
          1 - AMAG*THETAP*DCOS(PSI0(J)-THETA) ))
300       CONTINUE
C***      END OF PART II A)
C***      BEGIN BOOKEEPING
          DO 400 J=1,NPART-F
          PSI0(J) = PSIM(J+F)
          UOLD(J) = UNEW(J+F)
400       CONTINUE
          DO 500 J=F,1,-1
          UOLD(NPART+1-J)=KPOD-OMEGA+(J-1)*TAU*ALPHA1*(
          1 (KPOD*KPOD-BETAZ0*BETAZ0*OMEGA*KPOD)*
          1(AMAG*DCOS(PS00(NPART-J+(N-1)*F)-THETA))
          1+BETAZ0*BETAZ0*KPOD*(AMAG*GROW*DSIN(PS00(NPART-J+(N-1)*F)-THETA
)
          1 - AMAG*THETAP*DCOS(PS00(NPART-J+(N-1)*F)-THETA) ))
          PSI0(NPART+1-J)=PS00(NPART-J+(N-1)*F)+(J-1)*TAU*(KPOD-OMEGA
          1 +UOLD(NPART+1-J) )/2.
500       CONTINUE
C***      END OF PART II B) BOOKEEPING
          AMAG = AD
          GROWTH(N-1) = GROW
          THETA = THETA + DFLOAT(F)*TAU*THETAP
          FEQD = OMEGA + THETAP
          FEQ(N-1) = FEQD
C***      REPEAT PHASE AND AMPLITUDE INCREMENT
C***      FOR NEXT TIME STEP
          IF( FLOAT(N)/10. - N/10 .EQ. 0.) THEN
          WRITE(6,*) A(N),TIME(N)
          WRITE(2,*) A(N),TIME(N)
          END IF
1000      CONTINUE
          GROWTH(NTIMES) = GROW
          FEQ(NTIMES) = OMEGA + THETAP
          WRITE(6,1001)
          WRITE(2,1001)
1001      FORMAT(' THE FINAL PHASES ARE:')
C***      WRITE(6,*)(PSIM(J),J=1,NPART)
C***      WRITE(2,*)(PSIM(J),J=1,NPART)
          CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
          CALL PWRIT(500,80,%REF(AB1),70,1,0,0)
          CALL PWRIT(500,48,%REF(AB2),72,1,0,0)
          CALL PWRIT(500,16,%REF(AB3),75,1,0,0)
          CALL EZXY(TIME,A,NTIMES,%REF(GLAB))
          YLAB = ' GROWTH RATE $'
          GLAB = ' GROWTH RATE NORMALIZED TO TRANSIT TIME$'
          CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
          CALL EZXY(TIME,GROWTH,NTIMES,%REF(GLAB) )
          YLAB = ' REAL FREQUENCY $'
          GLAB = ' REAL FREQUENCY VS. TIME $'
          CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
          CALL EZXY(TIME,FEQ,NTIMES,%REF(GLAB))
```

```
1200      STOP
1210      END
```

```
MMMMMMMMMM    22222222222222222222222222222222222222222222222222    MMMMMMMMMM
MMMMMMMMMM    Digital Equipment Corporation - VAX/VMS Version V4.6    MMMMMMMMMM
MMMMMMMMMM    22222222222222222222222222222222222222222222222222    MMMMMMMMMM

                    M   M   AAA   RRRR    AAA   BBBB   L      EEEEE
                    MM MM  A   A  R   R  A   A  B   B  L      E
                    M M M  A   A  R   R  A   A  B   B  L      E
                    M   M  A   A  RRRR   A   A  BBBB   L      EEEE
                    M   M  AAAAA  R R    AAAAA  B   B  L      E
                    M   M  A   A  R  R   A   A  B   B  L      E
                    M   M  A   A  R   R  A   A  BBBB   LLLLL  EEEEE

TTTTTTTTTT  WW      WW    OOOOOO    FFFFFFFFFF    IIIIII    NN      NN
TTTTTTTTTT  WW      WW    OOOOOO    FFFFFFFFFF    IIIIII    NN      NN
    TT      WW      WW  OO      OO  FF              II      NN      NN
    TT      WW      WW  OO      OO  FF              II      NN      NN
    TT      WW      WW  OO      OO  FF              II      NNNN    NN
    TT      WW      WW  OO      OO  FF              II      NNNN    NN
    TT      WW      WW  OO      OO  FFFFFFFF        II      NN  NN  NN
    TT      WW      WW  OO      OO  FFFFFFFF        II      NN  NN  NN
    TT      WW  WW  WW  OO      OO  FF              II      NN    NNNN
    TT      WW  WW  WW  OO      OO  FF              II      NN    NNNN
    TT      WWWW  WWWW  OO      OO  FF              II      NN      NN
    TT      WWWW  WWWW  OO      OO  FF              II      NN      NN
    TT      WW      WW    OOOOOO    FF            IIIIII    NN      NN
    TT      WW      WW    OOOOOO    FF            IIIIII    NN      NN

            FFFFFFFFFF    OOOOOO    RRRRRRRR        ;;;;      222222
            FFFFFFFFFF    OOOOOO    RRRRRRRR        ;;;;      222222
            FF          OO      OO  RR      RR      ;;;;    22      22
            FF          OO      OO  RR      RR      ;;;;    22      22
            FF          OO      OO  RR      RR                      22
            FF          OO      OO  RR      RR                      22
            FFFFFFFF    OO      OO  RRRRRRRR        ;;;;          22
            FFFFFFFF    OO      OO  RRRRRRRR        ;;;;          22
            FF          OO      OO  RR  RR          ;;;;        22
            FF          OO      OO  RR  RR          ;;;;        22
    ....    FF          OO      OO  RR    RR          ;;      22
    ....    FF          OO      OO  RR    RR          ;;      22
    ....    FF            OOOOOO    RR      RR      ;;      2222222222
    ....    FF            OOOOOO    RR      RR      ;;      2222222222
```

File _VC$DRB1:[MARABLE.TWSTAG]TWOFIN.FOR;2 (4155,1,0), last revised on 17-JAN-1985 11:46, is a 11 block sequential file owned by UIC [MARABLE]. The records are variable length with a fixed control size of 2 bytes and implied (CR) carriage control. The longest record is 69 bytes.

Job TWOFIN (1986) queued to LN03_QUE on 21-MAR-1988 13:48 by user MARABLE, UIC [MARABLE], under account 4790 at priority 100, started on printer LTA8: on 21-MAR-1988 13:48 from queue VC_LN03B.


```
MMMMMMMMMM    22222222222222222222222222222222222222222222222222    MMMMMMMMMM
MMMMMMMMMM    Digital Equipment Corporation - VAX/VMS Version V4.6    MMMMMMMMMM
MMMMMMMMMM    22222222222222222222222222222222222222222222222222    MMMMMMMMMM
```

```
C***    TEST PROGRAM FOR MODEL EQUATIONS IN HIGH POWER
C***    FEL PROBLEM.
C***    REVISION 12/3 TO CORRECT BOUNDARY CONDITIONS
C***    REVISION 1/16 TO INCLUDE PHENOMENOLOGICAL DAMPING
        REAL A(3000),PSIM(4000),PSI0(4000),EPS,BETA1,TAU,DUM,OMEGA
        REAL TIME(3000),KPOD,UOLD(4000),UNEW(4000),GAM(4000),DUM1
        REAL GROWTH(3000),KWIGR,KWIGL,NU,RISE,GROW2(3000)
        INTEGER J,NPART,NTIMES,N,F
        PARAMETER (PI=3.141592653589)
        CHARACTER*40 GLAB,XLAB,YLAB
        CHARACTER*80 AB1,AB2,AB3,AB4
        PS00(J) = -OMEGA*J*TAU
        OPEN(UNIT=2,NAME='OUT.DAT',STATUS='NEW')
1       FORMAT('WAVE AMPLITUDE W/ REFL=',F5.3)
        XLAB =' TIME$'
        YLAB = ' WAVE AMPLITUDE$'
        WRITE(6,101)
101     FORMAT(' INPUT NO. OF PART.,NO. OF INTERAT.,GAM,BETAWIG,
        1 KWIGr,BUDKER,NWIG,EPS,F,RISE,NPLUS,REF ')
        READ(5,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,F,RISE,NPLUS,
REF
        WRITE(6,103)
        WRITE(2,103)
103     FORMAT(' INPUT DATA:NPART,NTIMES,GAM,BETAW,KWIGR,NU,NWIG
        1,EPS,F,RISE,NPLUS,REF IS:')
        WRITE(6,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,F,RISE,NPLUS
 REF
        WRITE(2,*)NPART,NTIMES,GAM0,BETAW,KWIGR,NU,NWIG,EPS,F,RISE,NPLUS
,REF
        KWIGL = 2.*FLOAT(NWIG)*PI
        BETA0 = SQRT(1. - 1./(GAM0*GAM0))
        BETAZ0 = SQRT(BETA0*BETA0 - BETAW*BETAW/2.)
        NOPT = JINT(2.*FLOAT(NWIG)*BETAZ0/(1.-BETAZ0)) +NPLUS
        OMEGA = (FLOAT(NOPT)*PI)/BETAZ0
        KPOD = KWIGL + BETAZ0*OMEGA
        WRITE(AB4,1)REF
        WRITE(AB1,104)NPART,NTIMES,GAM0,BETAW,NOPT
104     FORMAT('NPART=',I4,2X,'NTIMES=',I4,2X,'GAM=',F8.4,2X,
        1 'BETAWIG=',F8.4,2X,'NOPT=',I4)
105     FORMAT('KWIGR=',F8.4,3X,'NU=',E10.4,3X,'KPOD=',E10.4
        1,3X,'OMEGA=',E10.4)
        WRITE(AB2,105)KWIGR,NU,KPOD,OMEGA
        WRITE(AB3,106)KWIGL,EPS,F,RISE,NWIG,NPLUS
106     FORMAT('KWIGL=',F8.4,2X,'EPS=',E10.4,2X,'F=',I3,2X,
        1'RISE=',F8.4,2X,'NWIG=',I3,1X,'NP=',I3)
        BETA1 = 4.*NU*BETAW*(KWIGL)**2/(BETAZ0*BETA0*GAM0*OMEGA
        1 *(KWIGR*KWIGR))
        ALPHA1 = -BETAW/(2.*BETAZ0*BETAZ0)
        ALPHAR2 = (1. - REF)/BETAZ0
        ALPHAI2 =-4.*NU*KWIGL**2*(1.-BETAW**2/2.)/(BETAZ0**2*KWIGR**2
        1 *GAM0*OMEGA)
C***    INITIALIZE PHASE AND AMPLITUDE
        ARPRIM = 0.
        AIMPRIM = 0.
        A(1) = EPS
        TIME(1) = 1.
        TAU =1./FLOAT(NPART-1)
        TAUT = FLOAT(F)*TAU
        ARN = EPS*COS(3.*TAUT)
        AIMN = EPS*SIN(3.*TAUT)
        DO 100 J=1,NPART
        PSI0(J) = PS00(J-1) + (KPOD - OMEGA)*(NPART - J)*TAU
        UOLD(J) = KPOD - OMEGA
100     CONTINUE
        WRITE(6,102)
        WRITE(2,102)
```

```
102     FORMAT(' THE WAVE AMPLITUDES ARE: ')
C***    BEGIN LOOP FOR TIME INCREMENTS
        GROWTH(1) = 0.
        GROW2(1) = 0.
        DO 1000 N =2,NTIMES
        TIME(N) = 1. + FLOAT(N-1)*TAUT
        TRISE = (1. -EXP( (1.-TIME(N))/RISE ))
C***    BEGIN PART II: STEP AMPLITUDES AND PHASES
C***    COMPLETE SUM FOR AMPLITUDE STEP
        DUM = 0.
        DUM1 = 0.
        DO 200 J=1,NPART
        BETAZ = BETAZ0*(UOLD(J) + OMEGA)/KPOD
        GAM(J)= SQRT( (1.-BETAZ*BETAZ)/(1./(GAM0*GAM0)+BETAW*BETAW/2.) )
        DUM = DUM + COS(PSI0(J))*TAU*GAM(J)
        DUM1 = DUM1 + SIN(PSI0(J))*TAU*GAM(J)
200     CONTINUE
        ARN1 = ARN +TAUT*((ALPHAI2*AIMN+BETA1*DUM)*TRISE-ALPHAR2*ARN)
        AIMN1=AIMN +TAUT*((-ALPHAI2*ARN+BETA1*DUM1)*TRISE-ALPHAR2*AIMN)
        A(N) = SQRT(ARN1*ARN1 + AIMN1*AIMN1)
        DO 300 J=1,NPART
        PSIM(J) = PSI0(J) + TAUT*UOLD(J)
        UNEW(J) = UOLD(J) + TAUT*ALPHA1*GAM(J)*GAM(J)*(
        1 (KPOD*KPOD-BETAZ0*BETAZ0*OMEGA*(UOLD(J)+OMEGA))*
        1 (ARN*COS(PSI0(J))+AIMN*SIN(PSI0(J))) +
        1 BETAZ0*BETAZ0*(UOLD(J)+OMEGA)*(ARPRIM*SIN(PSI0(J))
        1 - AIMPRIM*COS(PSI0(J)) ))
300     CONTINUE
C***    END OF PART II A)
C***    BEGIN BOOKEEPING
        DO 400 J=1,NPART-F
        PSI0(J) = PSIM(J+F)
        UOLD(J) = UNEW(J+F)
400     CONTINUE
        DO 500 J=F,1,-1
        UOLD(NPART+1-J)=KPOD-OMEGA+(J-1)*TAU*ALPHA1*(
        1 (KPOD*KPOD-BETAZ0*BETAZ0*OMEGA*KPOD)*
        1(ARN*COS(PS00(NPART-J+(N-1)*F))+AIMN*SIN(PS00(NPART-J+(N-1)*F))
)
        1 +BETAZ0*BETAZ0*KPOD*(ARPRIM*SIN(PS00(NPART-J+(N-1)*F))
        1 - AIMPRIM*COS(PS00(NPART-J+(N-1)*F))))
        PSI0(NPART+1-J)=PS00(NPART-J+(N-1)*F)+(J-1)*TAU*(KPOD-OMEGA
        1 +UOLD(NPART+1-J) )/2.
500     CONTINUE
C***    END OF PART II B) BOOKEEPING
        ARPRIM = (ALPHAI2*AIMN+BETA1*DUM)*TRISE-ALPHAR2*ARN
        AIMPRIM = (-ALPHAI2*ARN+BETA1*DUM1)*TRISE-ALPHAR2*AIMN
        GROWTH(N) =(ARN1*ARPRIM + AIMN1*AIMPRIM)/(A(N)*A(N))
        GROW2(N) = (A(N) - A(N-1))/(A(N-1)*TAUT)
        ARN = ARN1
        AIMN = AIMN1
C***    REPEAT PHASE AND AMPLITUDE INCREMENT
C***    FOR NEXT TIME STEP
        IF( FLOAT(N)/10. - N/10 .EQ. 0.) THEN
        WRITE(6,*) A(N),TIME(N)
        WRITE(2,*) A(N),TIME(N)
        END IF
1000    CONTINUE
        WRITE(6,1001)
        WRITE(2,1001)
1001    FORMAT(' THE FINAL PHASES ARE:')
C***    WRITE(6,*)(PSIM(J),J=1,NPART)
C***    WRITE(2,*)(PSIM(J),J=1,NPART)
        CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
        CALL PWRIT(500,80,%REF(AB1),70,1,0,0)
        CALL PWRIT(500,48,%REF(AB2),72,1,0,0)
```

```
      CALL PWRIT(500,16, %REF(AB3),75,1,0,0)
      CALL EZXY(TIME,A,NTIMES,%REF(AB4))
      YLAB = ' GROWTH RATE $'
      GLAB = ' GROWTH RATE NORMALIZED TO TRANSIT TIME$'
      CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
      CALL EZXY(TIME,GROWTH,NTIMES,%REF(GLAB) )
      GLAB = 'GROWTH RATE EVALUATED BY DERIV.$'
      CALL ANOTAT(%REF(XLAB),%REF(YLAB),1,0,0,DSHL)
      CALL EZXY(TIME,GROW2,NTIMES,%REF(GLAB))
      STOP
      END
```

B. NRL MEMORANDUM REPORT 5679

CONTENTS

HIGH GAIN FREE ELECTRON LASER OSCILLATORS

I. Introduction

We have conducted an analytical and numerical analysis of the field evolution in a high gain free electron laser operating in the oscillator configuration, as depicted in Fig. 1. The analysis is applicable to systems with electron beam pulse lengths which are longer that the particle transit time in the resonator. The electron beam equilibrium is therefore assumed to be spatially uniform and temporally stationary. The radiation field and phase averages which are performed with the ensemble of electrons is conducted for an interaction length which consists of the entire wiggler structure. This is in contrast to other simulations (theories) which perform the ensemble average over the wavelength of the ponderomotive potential; as is applicable to systems with temporally stationary fields[1] or short beam pulses[2] that are spatially periodic.

We find that the numerical simulations yield qualitative and quantitative agreement with the theory. The theory for the example given (strong pump Compton regime) can be separated into three operation regimes which we shall denote as the ultra-high gain, moderate gain and low gain regimes. Both the ultra-high gain ($\Gamma_k L >> 1$) and the low gain ($\Gamma_k L << 1$) regimes yield growth rates that exhibit the same scaling with beam current, energy and wiggler field as is obtained for an FEL amplifier operating in these regimes. Additionally, we consider a moderate gain regime ($\Gamma_k L \geq 1$) which is of direct interest to NRL experimental parameters.

II. Theoretical Model

An analysis of the space time evolution of the fields and particles within an FEL oscillator requires a self-consistent coupling of the fundamental equations for the particles

Manuscript approved December 6, 1985.

and fields[3–8]. We have considered a Maxwell-Vlasov description of the fields and particles. The analysis in Appendix A results in the following system of equations for the coupling of the fields and particles. The backward travelling wave evolves according to, $\partial\tilde{a}_b(z)/\partial z - i\alpha\tilde{a}_b(z) = 0$. The forward travelling potential and the electrostatic potential evolve according to,

$$\frac{\partial}{\partial z}\tilde{a}_f(z) + i\alpha\tilde{a}_f(z) =$$
$$-c_1\int_0^z dz'(z'-z)\exp[-i\Delta K(z'-z)](\partial/\partial z' - iK)[\beta_w\tilde{a}_f(z')/2 - \tilde{\phi}(z')]$$
$$+ic_2\int_0^z dz'\exp[-i\Delta K(z'-z)](\partial/\partial z' - iK)[(1+\beta_{z0}^2)\beta_w\tilde{a}_f(z')/2 - \tilde{\phi}(z')], \quad (1)$$
$$\frac{\partial}{\partial z}\tilde{\phi}(z) - iK\tilde{\phi}(z)/2 =$$
$$-c_3\int_0^z dz'(z'-z)\exp[-i\Delta K(z'-z)](\partial/\partial z' - iK)[\beta_w\tilde{a}_f(z')/2 - \tilde{\phi}(z')]$$
$$+ic_4\int_0^z dz'\exp[-i\Delta K(z'-z)](\partial/\partial z' - iK)[\beta_w\tilde{a}_f(z')/2 - (1-\beta_{z0}^2)\tilde{\phi}(z')]. \quad (2)$$

The parameters in Eqs. (1) and (2) are given by, $\alpha = \Delta\omega/c - \omega_p^2(1-\beta_w^2)/2\omega_0 c\overline{\gamma}$, $c_1 = c_2(1-\beta_{z0}^2)(\omega_0+\Delta\omega)/v_{z0}$, $c_2 = \omega_p^2\beta_w/2\omega_0 c\beta_{z0}^2\overline{\gamma}$, $c_3 = c_1\omega_0/\beta_w Kc$, $c_4 = c_3/(1-\beta_{z0}^2)$, $K = k_0 + k_w$ and $\Delta K = K - (\omega_0+\Delta\omega)/v_{z0}$. By making use of the convolution theorem, the Laplace transform of Eqs. (1) and (2) yields,

$$\{(s+i\alpha)[(s-i\Delta K)^2 + 2c_3] + iKc_1\beta_w/2\}\hat{a}_f(s) =$$
$$\{(s-i\Delta K)^2 + 2c_3(1-\beta_w^2/4)\}\tilde{a}_f(0), \quad (3)$$
$$\{(s-i\Delta K)^2 + 2c_3\}\hat{\phi}(s) = c_3\beta_w[\hat{a}_f(s) - i\tilde{a}_f(0)/K], \quad (4)$$

where $\hat{a}_f(s)$ and $\hat{\phi}(s)$ are the Laplace transformed vector and scalar potentials, and we have retained only the terms in the driving current that arise from the momentum derivatives of the phase, $\partial\Delta K/\partial p_z$. We have also assumed that the electron beam enters the resonator unbunched so that $\tilde{\phi}(z=0) = 0$.

Since the singularities of the Laplace transformed potentials are isolated poles, the Bromwich inversion of these transforms can be easily performed,

$$\tilde{\phi}(z) = \sum_j \text{Residue}\{\hat{\phi}(s), s_j\} \exp(s_j z), \tag{5}$$

$$\tilde{a}_f(z) = \sum_j \text{Residue}\{\hat{a}_f(s), s_j\} \exp(s_j z), \tag{6}$$

where s_j, are the poles of the Laplace transformed potentials. The solution for the radiation potential given in Eq. (6) is of the same generic form as the solution obtained by Bernstein and Hirshfield[9] for the FEL amplifier configuration. Our analysis shall differ in that the backward travelling waves are not neglected, and the combination of the forward and backward travelling waves are required to satisfy the appropriate boundary conditions at the mirror surfaces. Specifically, the tangential component of the electric field must be zero at the non-transmitting mirror surface, i.e., $\tilde{a}_f(0) + \tilde{a}_b(0) = 0$. At the partially transmitting mirror surface at the end of the resonator, the tangential components of the electric field must be continuous, i.e., $\tilde{a}_f(L)\exp(-ik_0L) + \tilde{a}_b(L)\exp(ik_0L) - (1 - \sqrt{R})\tilde{a}_f(L)\exp(-ik_0L) = 0$, where L is the length of the resonator and R is the fractional power reflected from the far end resonator mirror. This yields the following expression for the boundary conditions at the mirror surfaces,

$$\exp\frac{i\Delta\omega L}{c} = \frac{\sqrt{R}}{\tilde{a}_f(0)} \exp\{i[\frac{\omega_p^2}{2\omega_0 c\bar{\gamma}}(1-\beta_w^2/2) - 2k_0]L\} \sum_j \text{Residue}\{\hat{a}_f(s), s_j\} \exp(s_j L), \tag{7}$$

which is the equation that self-consistently determines the complex operating frequency, $\Delta\omega$, of the oscillator.

III. Results for Compton Regime

In the Compton regime the effect of the electrostatic potential can be neglected[10,11]. In addition we shall assume that the spatial derivative of the vector potential in the driving current is negligible. These derivatives are negligible when, $|(s - i\Delta K)^2| >> 2c_3$ and $|s| << K$. Under these conditions the Laplace transform of the vector potential is given by,

$$\hat{a}_f(s) = \tilde{a}_f(0)(s - i\Delta K)^2 / \prod_{j=1}^{3}(s - s_j), \tag{8}$$

where s_j are the roots of the dispersion relation, $(s - i\Delta K)^2(s + i\alpha) = -ic_1\beta_w K/2 \approx -i\omega_p^2(1+\beta_{z0})\beta_w^2 k_w / 4\beta_{z0}^3 c^2\overline{\gamma}$. Since k_0 is a free parameter, choose k_0 such that, $\Delta K = k_0 + k_w - (\omega_0 + \Delta\omega)/v_{z0} = -\Delta\omega/c + \omega_p^2(1 - \beta_w^2/2)/2\omega_0 c\overline{\gamma}$. Which results in the following solutions to the dispersion relation,

$$s_j = -i\left[\frac{\Delta\omega}{c} - \frac{\omega_p^2}{2\omega_0 c\overline{\gamma}}(1 - \beta_w^2/2)\right] + \Gamma_0 \begin{cases} i2/\sqrt{3} \\ 1 - i/\sqrt{3} \\ -1 - i/\sqrt{3} \end{cases}, \tag{9}$$

where $\Gamma_0 = \sqrt{3}k_w[\omega_p^2(1+\beta_{z0})\beta_w^2/4k_w^2 c^2\overline{\gamma}]^{1/3}/2\beta_{z0}$ is a spatial growth rate corresponding to the largest spatial growth rate in the amplifier case[12]. By evaluating the residues of $\hat{a}_f(s)$ for each of these poles, one obtains the following solution for the spatial structure of the radiation potential,

$$\tilde{a}_f(z) = \frac{\tilde{a}_f(0)}{3}\sum_{j=1}^{3}\exp(s_j z). \tag{10}$$

It is evident from Eq.(10) that the spatial growth of the radiation field can be described by the interference of three modes; which can be identified as the positive and negative energy beam modes, and a transverse electromagnetic mode. The constructive or destructive nature of this interference is dependent on the values of the physical parameters which characterize the roots, s_j. For physical parameters such that, $\Gamma_0 L >> 1$, the unstable mode dominates and one obtains exponential spatial growth at the rate Γ_0.

The temporal growth rate of the radiation field is obtained from the negative imaginary part of the complex oscillator frequency, $\Delta\omega$. The oscillator frequency is determined by the boundary conditions as expressed in Eq. (7), which for the approximate roots under consideration yields,

$$\exp\frac{i\Delta\omega L}{c} = \frac{\sqrt{R}}{3}\exp\left\{i\left[\frac{\omega_p^2}{2\omega_0 c\overline{\gamma}}(1 - \beta_w^2/2) - 2k_0\right]L\right\}\sum_{j=1}^{3}\exp(s_j L). \tag{11}$$

We shall consider three distinct solutions to this equation for the complex oscillator frequency. The first of which is the ultra-high gain regime ($\Gamma_0 L >> 1$), in which case, only the fastest growing mode in Eq. (11) is retained. The second case is the moderate gain regime ($\Gamma_0 L \geq 1$), where only the decaying mode in Eq. (10) is neglected. The final case is valid for arbitrary gain and all terms in Eq. (11) are retained. The imaginary part of the oscillator frequency yields the following temporal growth rates,

$$\Gamma_\omega \frac{L}{c} = \frac{1}{2} \ln \frac{\sqrt{R}}{3} + \Gamma_0 L/2, \qquad \text{Ultra-High Gain} \qquad (12)$$

$$\Gamma_\omega \frac{L}{c} = \frac{1}{2} \ln \frac{\sqrt{R}}{3} + \frac{1}{4} \ln[2\cosh(\Gamma_0 L) + 2\cos(\sqrt{3}\Gamma_0 L)], \qquad \text{Moderate Gain} \qquad (13)$$

$$\Gamma_\omega \frac{L}{c} = \frac{1}{2} \ln \frac{\sqrt{R}}{3} + \frac{1}{4} \ln[1 + 4\cos(\sqrt{3}\Gamma_0 L)\cosh(\Gamma_0 L) + 4\cosh^2(\Gamma_0 L)]. \qquad \text{Arbitrary Gain} \qquad (14)$$

In each of the expressions for the growth rate the first term is negative definite. This represents the effect of losses at the mirrors and the coupling losses due to the splitting of the radiation into three modes. The necessary condition for the oscillator to lase is that the remaining terms exceed this loss. For the ultra-high gain case this requires $\Gamma_0 L \geq -\ln(\sqrt{R}/3)$. This expression has been confirmed experimentally[13] in recent operation of the NRL FEL oscillator. The interaction length, L, can be varied by dumping the beam at different axial locations within the wiggler. For the following set of experimental parameters, beam energy $E_0 = 500\ keV$, beam current $I = 100\ A$, wiggler field strength $B_w = 615\ G$, beam radius $r_b = 0.64\ cm$ and wiggler length $\ell_w = 4.0\ cm$, the minimum interaction length is determined to be 45 cm. Inserting this value into Eq.(12) yields a theoretical value of 0.64 for the reflection coefficient. The independently measured Bragg reflection coefficient has the value 0.65, which is in excellent agreement with the theoretical value.

IV. Multi-Mode Simulation

The space-time evolution of the fields in the resonator is simulated by numerically evolving the equations for the fields and particles. The radiation field model for the multi-mode simulation is given by,

$$\vec{A}_R(z,t) = \sum_n a_n(t)\sin(k_n z)\exp(i\omega_n t)\hat{e}_- + c.c., \tag{15}$$

$$\phi(z,t) = \sum_n \phi_{1n}(t)\sin[(k_n + k_w)z - \omega_n t] + \phi_{2n}(t)\cos[(k_n + k_w)z - \omega_n t], \tag{16}$$

where $k_n = n\pi/L = \omega_n/c$ and the sum is over the discrete number of modes under consideration. This model has the property that the complex expansion coefficients in the harmonic analysis, $a_n(t), \phi_{1n}(t), \phi_{2n}(t)$, are only functions of time, which results in ordinary differential equations for the particle and field evolution. This model also has the attribute that the field boundary conditions at two perfectly reflecting mirrors is automatically satisfied, and we model the resonator losses heuristically by adding a damping term to the wave equation, $[\partial^2/\partial z^2 - c^{-2}\partial^2/\partial t^2 - \nu c^{-2}\partial/\partial t]\vec{A}_R(z,t) = 4\pi c^{-1}\vec{J}_\perp(z,t)$, where $\nu = \omega/Q$ and Q is the quality factor of the resonator. The driving currents and charge densities for the vector and scalar potentials are modeled with a discrete distribution function as follows,

$$\rho(z,t) = -e\int dz_0 n_0(t)\delta(z - \tilde{z}(z_0,t)), \tag{17}$$

$$\vec{J}_\perp(z,t) = -\frac{e^2\vec{A}}{mc}\int dz_0 n_0(t)\delta(z - \tilde{z}(z_0,t))/\gamma_0, \tag{18}$$

where, $\vec{A} = \vec{A}_R + \vec{A}_w$, $n_0(t) = n_0(\infty)[1 - \exp(-t/t_R)]$ and $n_0(\infty)$ is the flattop density of the electron beam pulse, t_R is the characteristic rise time for the beam current or density, and $\tilde{z}(z_0,t)$ is the axial orbit of a particle located at position z_0 at $t = 0$.

The slowly varying field approximation, $\partial a_n(t)/\partial t << \omega_n a_n(t)$, yields the following set of equations for the evolution of the fields and particles.

$$\hat{A}'_n = -\frac{\nu^* \hat{A}_n}{2} - \beta_{1n} \int_{\tau-1}^{\tau} d\tau_0 \sin[\tilde{\psi}_n(\tau_0, \tau)] \frac{\overline{\gamma}}{\gamma_0} P(\tau_0), \tag{19}$$

$$\theta'_n = -\frac{\nu_I^*}{2} + \frac{\beta_{1n}}{\hat{A}_n} \int_{\tau-1}^{\tau} d\tau_0 \cos[\tilde{\psi}_n(\tau_0, \tau)] \frac{\overline{\gamma}}{\gamma_0} P(\tau_0), \tag{20}$$

$$\hat{\phi}_{1n} = -\beta_{2n} \int_{\tau-1}^{\tau} d\tau_0 \cos[\tilde{\psi}_n(\tau_0, \tau)] P(\tau_0), \tag{21}$$

$$\hat{\phi}_{2n} = -\beta_{2n} \int_{\tau-1}^{\tau} d\tau_0 \sin[\tilde{\psi}_n(\tau_0, \tau)] P(\tau_0), \tag{22}$$

$$\begin{aligned} \tilde{\psi}''_n = &-\theta''_n + \beta_{3n} \sum_m (k_m + k_w) L [\hat{\phi}_{2m} \cos[\tilde{\psi}_m(\tau_0, \tau)] - \hat{\phi}_{1m} \sin[\tilde{\psi}_m(\tau_0, \tau)]] \\ &+ \beta_{4n} \sum_m \Big\{ [(k_m + k_w)L - k_m L \beta_z + \beta_{0z}\beta_z \theta'_m] \hat{A}_m \sin[\tilde{\psi}_m(\tau_0, \tau)] \\ &\qquad - \beta_{0z}\beta_z \hat{A}'_m \cos[\tilde{\psi}_m(\tau_0, \tau)] \Big\}. \end{aligned} \tag{23}$$

We have introduced the following normalized parameters, $\tau = v_{0z}t/L$, $\hat{A}_n = eA_n/mc^2$, $\hat{\phi} = e\phi/mc^2$, $(...)' = \partial(...)/\partial\tau$. We have also made the following definitions, $\hat{a}_n = -i\hat{A}_n(\tau)\exp(i\theta_n\tau)$, $\tilde{\psi}_n(\tau_0, \tau) = (k_n + k_w)\tilde{z}(\tau_0, \tau) - \omega_n L\tau/v_{0z} - \theta_n(\tau)$, $\nu_R^* = \nu L/v_{0z}$, $\nu_I^* = \omega_p^2 k_w L F(1 - \beta_w^2/2)/2k_w^2 c^2 \overline{\gamma}$, $\beta_{1n} = F\beta_w L \beta_0 \overline{\gamma} \omega_p^2 / 2c^2 \beta_{0z}^2 k_n$, $\beta_{2n} = 2F\omega_p^2 \beta_0 / c^2 (k_n + k_w)^2 \beta_{0z}$, $\beta_{3n} = L(k_n + k_w)/\overline{\gamma}\gamma_z^2 \beta_{0z}^2$, $\beta_{4n} = \beta_w (k_n + k_w) L \overline{\gamma} / 2\beta_{0z}^2 \gamma_0^2$ and $P(\tau_0) = 1 - \exp(-\tau_0/\tau_R)$, where L is the length of the resonator, F is the filling factor and ω_p is the nonrelativistic plasma frequency.

This system of equations is solved numerically by using a four-point Adams–Bashforth predictor corrector scheme which is initialized by using the three point Runge–Kutta method. The ensemble average over initial electrons, $\int_{\tau-1}^{\tau}(...)d\tau_0$, is typically performed with two thousand (2000) particles. The results of the simulations and the linear theory, obtained from the linearization of Eqs. (19) - (23), are shown in Figs. 2 and 3.

V. Conclusions

A comparison of the temporal growth rates obtained from the linear theory and the numerical simulations is shown in Figs. 2 and 3. The growth rates for the simulations are obtained numerically from the field amplitude data during the initial field evolution, where

the wave growth is linear. In Fig. 2 the data is presented for a low gain case with physical parameters given by, $\gamma = 2.0, I = 5A, F = 0.2, k_w r_b = 0.62831, \beta_w = 0.2$ and $L/\ell_w = 50$. There is an excellent agreement between theory and simulation. In Fig. 2 we also compare the theoretical and numerical efficiencies for a high gain case. Where the efficiency is defined as the stored electromagnetic energy density normalized to the incident beam energy density. The theoretical estimates of the efficiency are based on particle trapping arguments[12], with the assumption that all the energy lost by the particles is converted into electromagnetic energy. The characteristic change in velocity of a particle is given by the difference in the beam velocity and the phase velocity of the trapping potential. This phase velocity is approximated from the results of the linear dispersion relation. Again we find good qualitative and quantitative agreement between the simulation and theory.

Acknowledgment

We gratefully acknowledge many fruitful discussions and experimental data provided by Drs. John Pasour and Joe Mathews. This work was supported by DARPA under contract # 5483.

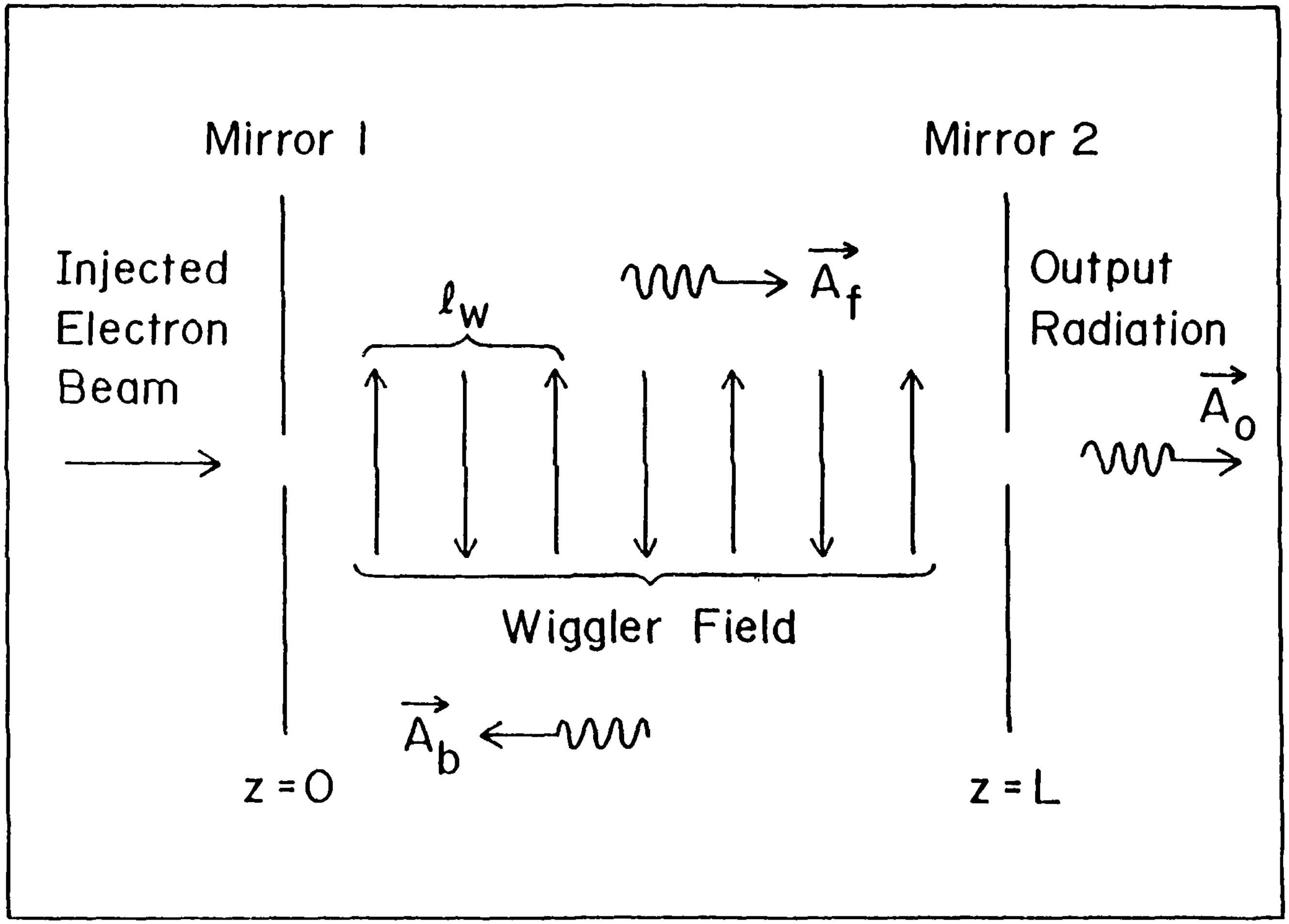


Figure 1. Schematic of high gain FEL oscillator, where $\vec{A}_f = a_f(z,t)\exp(-ik_0z + i\omega_0 t)\hat{e}_- +$ $c.c.$ is the forward propagating wave, $\vec{A}_b = a_b(z,t)\exp(ik_0z + i\omega_0 t)\hat{e}_- + c.c$ is the backward propagating wave and $\vec{A}_0 = (1-\sqrt{R})\vec{A}_f$ is the transmitted wave.

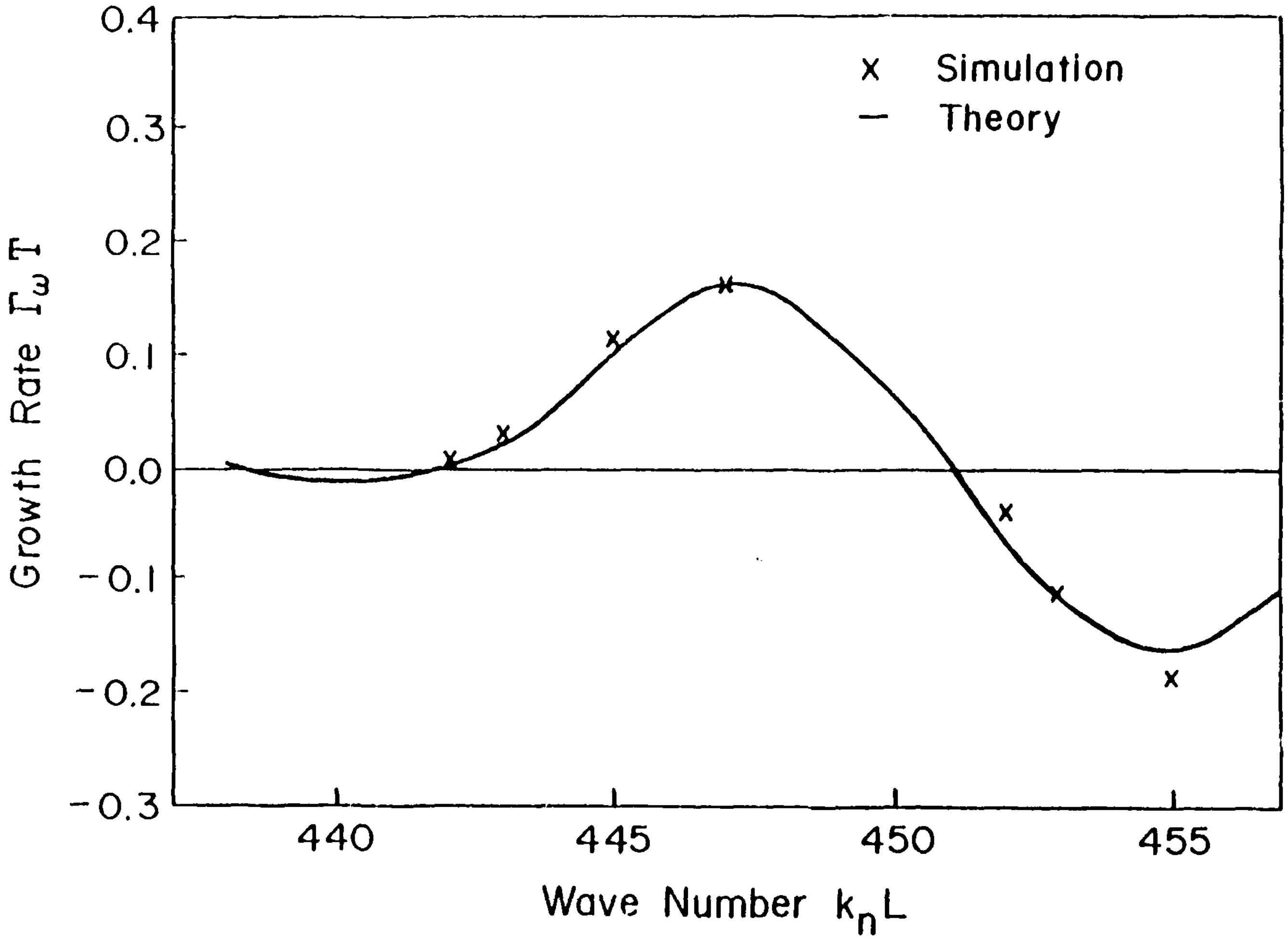


Figure 2. Comparison of the temporal growth rate from theory and numerical simulation in low gain Compton regime. The following physical parameters are used: $\gamma = 2.0$, $I = 5A$, $F = 0.2$, $k_w r_b = 0.62831$, $\beta_w = 0.2$ and $L/\ell_w = 50$.

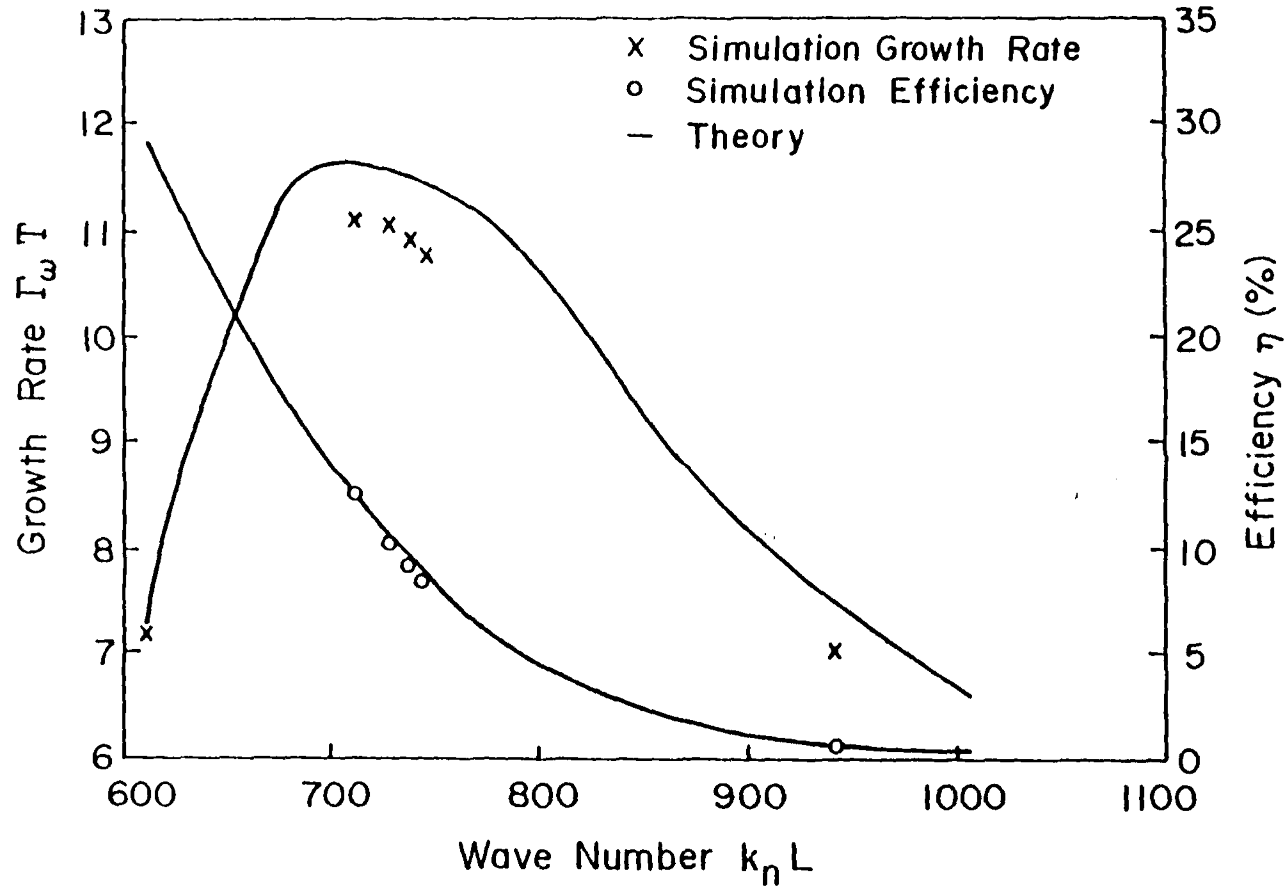


Figure 3. Comparison of the temporal growth rate and saturation efficiency from theory and numerical simulations in high gain Compton regime. The following physical parameters are used: $\gamma = 2.5$, $I = 500A$, $F = 0.2$, $k_w r_b = 0.7853$, $\beta_w = 0.3686$ and $L/\ell_w = 22$.

References

1. P. Sprangle, C.M. Tang and W.M. Manheimer, Phys. Rev.A 21, 302 (1980). ADA114220
2. C.M. Tang and P. Sprangle, NRL Memo Report 4774 (1982).
3. *Free-Electron Generators of Coherent Radiation, Physics of Quantum Electronics* vols. 7–9, eds. S.F. Jacobs, H.S. Pillof, M. Sargent III, M.O. Scully and R. Spitzer (Addison-Wesley, New York) 1980.
4. W.B. Colson, Phys. Lett. 64A, 1 190 .977)
5. R.C. Davidson and H.S. Uhm, J. Appl. Phys. 53 ,2910 (1982).
6. H.P. Freund and P. Sprangle, Phys. Rev. A26, 2004 (1982).
7. IEEE J. Quantum Electronics QE-17 (1981).
8. IEEE J. Quantum Electronics QE-19 (1983).
9. I. Bernstein and J. Hirshfield, Phys. Rev. Lett.40, 761 (1978).
10. P. Sprangle and T. Coffey, Physics Today 37, 44 (1984)
11. N.M. Kroll and W. A. McMullin, Phys. Rev. A17,300 (1978).
12. P. Sprangle, R.A. Smith and V.L. Granatstein, *Infrared and Millimeter Waves*, ed. by K. Button (Academic, New York, 1979) p. 279 – Vol. 1.
13. J. Pasour and J. Mathews to be published.

Appendix A

In the following, we shall consider the space-time evolution of the radiation fields produced by the interaction of a beam of relativistic electrons with a helical wiggler field contained within the mirrors of an optical resonator. The analysis is fully relativistic and is conducted self-consistently within the framework of the Vlasov-Maxwell system of equations.

The wiggler vector potential is modeled as follows,

$$\vec{A}_w(z) = \frac{B_w}{k_w}\left[\exp(ik_w z)\hat{e}_- + \exp(-ik_w z)\hat{e}_+\right], \tag{A1}$$

where the wiggler magnetic field strength is B_w, the wiggler period is $\ell_w = 2\pi/k_w$ and the basis vectors are $\hat{e}_\pm = (\hat{e}_x \pm i\hat{e}_y)/2$. We have assumed in this model that the beam radius is small compared to the wiggler period $(k_w r_b < 1)$ hence the transverse gradients in the wiggler field are neglected. We similarly invoke the para-axial approximation to the radiation fields and neglect transverse coordinate dependencies in the fields, to obtain the following radiation field model for the vector and scalar potentials,

$$\vec{A}(z,t) = \left[a_f(z,t)\exp(-ik_0 z) + a_b(z,t)\exp(ik_0 z)\right]\exp(i\omega_0 t)\hat{e}_- + \text{c.c.}, \tag{A2}$$

$$\phi(z,t) = \tilde{\phi}(z,t)\exp[-i(k_0 + k_w)z + i\omega_0 t] + \text{ c.c.}, \tag{A3}$$

where $a_f(z,t)$ and $a_b(z,t)$ denote the forward and backward components of the wave field respectively, and $\omega_0 = ck_0$ is the frequency. These field coefficients are assumed to be slowly varying functions of space and time compared to the radiation wavelength and temporal period. The slow spatial dependence of the field coefficients is expressed by, $| Q^{-1}\partial Q/\partial z | << k_0$, with $Q = a_f(z,t), a_b(z,t), \tilde{\phi}(z,t)$ and the slow temporal dependence of the the field coefficients is expressed by, $| Q^{-1}\partial Q/\partial t | << \omega_0$.

The space-time evolution of the fields is governed by Maxwell's equations, which can be cast in the form, $(\partial^2/\partial z^2 - c^{-2}\partial^2/\partial t^2)\vec{A}(z,t) = 4\pi c^{-1}\vec{J}_\perp(z,t)$ and $\partial^2/\partial z^2\varphi(z,t) = -4\pi\rho(z,t)$. The driving current and charge densities are obtained from the appropriate moments of the Vlasov distribution function. The Vlasov distribution function is evolved

according to the equation, $\{\partial/\partial t + (p_z/m\gamma)\partial/\partial z - e[\vec{E} + (\vec{p}\times\vec{B})/m\gamma c]\cdot\partial/\partial\vec{p}\}g(z,\vec{p},t) = 0$. By making use of the fact that the canonical transverse momentum is an invariant of the motion and assuming that the beam is cold in the transverse direction (e.g., $g(z,P_x,P_y,p_z,t) = \tilde{g}(z,p_z,t)\delta(P_x)\delta(P_y)$) the evolution of the reduced distribution function is governed by,

$$\left\{\frac{\partial}{\partial t} + \frac{p_z}{m\gamma_T}\frac{\partial}{\partial z} + \left[e\frac{\partial\phi}{\partial z} - \frac{e^2}{2m\gamma_T c^2}\frac{\partial}{\partial z}(\vec{A}\cdot\vec{A})\right]\frac{\partial}{\partial p_z}\right\}\tilde{g}(z,p_z,t) = 0, \tag{A4}$$

where $mc^2\gamma_T = \{m^2c^4 + c^2p_z^2 + e^2(\vec{A}\cdot\vec{A})\}^{1/2}$. Since $\partial(\vec{A}_w\cdot\vec{A}_w)/\partial z = 0$, the equilibrium distribution function satisfies the equation, $\{\partial/\partial t+(p_z/m\gamma_0)\partial/\partial z\}\tilde{g}^{(0)}(z,p_z,t) = 0$, where $mc^2\gamma_0 = \{m^2c^4 + c^2p_z^2 + e^2B_w^2/k_w^2\}^{1/2}$. For long electron beam pulses we shall consider spatially and temporally homogeneous equilibria given by $\tilde{g}^{(0)}(z,p_z,t) = \tilde{g}^{(0)}(p_z)$. To first order in the perturbed fields, the evolution of the linearized distribution function is given by,

$$\left\{\frac{\partial}{\partial t} + \frac{p_z}{m\gamma_0}\frac{\partial}{\partial z}\right\}\tilde{g}^{(1)}(z,p_z,t) = \left[-e\frac{\partial\phi}{\partial z} + \frac{e^2}{m\gamma_0c^2}\frac{\partial}{\partial z}(\vec{A}\cdot\vec{A}_w)\right]\frac{\partial\tilde{g}^{(0)}}{\partial p_z}. \tag{A5}$$

The solution to the linearized Vlasov equation is formally given by,

$$\begin{aligned}\tilde{g}^{(1)}(z,p_z,t) = \int_0^z dz'\frac{m\gamma_0}{p_z}\Big[&-e\frac{\partial}{\partial z'}\phi(z',t+\frac{(z'-z)}{v_z}) \\ &+ \frac{e^2}{m\gamma_0c^2}\frac{\partial}{\partial z'}(\vec{A}_w(z')\cdot\vec{A}(z',t+\frac{(z'-z)}{v_z}))\Big]\frac{\partial\tilde{g}^{(0)}}{\partial p_z}.\end{aligned} \tag{A6}$$

Linearizing the wave equations for $\vec{A}$ and ϕ one obtains,

$$\begin{aligned}\left(\frac{\partial^2}{\partial t^2} - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right)\vec{A}(z,t) = \frac{-4\pi}{c}\Big\{&\frac{-e^2}{c}\vec{A}_w\int dp_z\frac{\tilde{g}^{(1)}(z,p_z,t)}{m\gamma_0} \\ &-\frac{e^2}{c}\vec{A}\int dp_z\frac{\tilde{g}^{(0)}(p_z)}{m\gamma_0} + \frac{e^2}{c}\vec{A}_w\left(\frac{e^2\vec{A}_w\cdot\vec{A}}{m^2c^4}\right)\int dp_z\frac{\tilde{g}^{(0)}(p_z)}{m\gamma_0^3}\Big\},\end{aligned} \tag{A7}$$

$$\frac{\partial^2}{\partial z^2}\phi(z,t) = 4\pi n_0 e\int dp_z\tilde{g}^{(1)}(z,p_z,t). \tag{A8}$$

By making use of the slowly varying coefficient approximations, the components of Maxwell's equations can be expressed as:

$$\left\{\frac{\partial}{\partial z}+\frac{1}{c}\frac{\partial}{\partial t}-\frac{i\omega_p^2}{2\omega_0 c\overline{\gamma}}\left(\alpha^{(1)}-\alpha^{(3)}\frac{\beta_w^2}{2}\right)\right\}a_f(z,t)=$$

$$+\frac{i\omega_p^2}{2\omega_0 c\overline{\gamma}}\frac{B_w}{k_w}\exp[-i(Kz-\omega_0 t)]\int dp_z\frac{\overline{\gamma}}{\gamma_0}\tilde{g}^{(1)}(z,p_z,t), \tag{A9}$$

$$\left\{\frac{\partial}{\partial z}-\frac{1}{c}\frac{\partial}{\partial t}+\frac{i\omega_p^2}{2\omega_0 c\overline{\gamma}}\left(\alpha^{(1)}-\alpha^{(3)}\frac{\beta_w}{2}\right)\right\}a_b(z,t)=0, \tag{A10}$$

$$\left\{\frac{\partial}{\partial z}-\frac{i}{2}K\right\}=i\frac{2\pi n_0 e}{K}\exp[i(Kz-\omega_0 t)]\int dp_z\tilde{g}^{(1)}(z,p_z,t), \tag{A11}$$

where $mc^2\overline{\gamma}=\{m^2c^4+c^2p_{z0}^2+e^2B_w^2/k_w^2\}^{1/2}$, $\alpha^{(n)}=\int dp_z(\overline{\gamma}/\gamma_0)^n\tilde{g}^{(0)}(p_z)$, $\omega_p^2=4\pi n_0 e^2/m$, $\beta_w=eB_w/m\overline{\gamma}c^2k_w$ and $K=k_0+k_w$. Also note that in the frequency regime for resonant interaction of the radiation field and the beam particles, ($\omega_0\approx 2\beta_z\gamma_z^2k_wc$) the driving current for the backward wave is negligible. We shall solve this system of equations in the time asymptotic limit for which the forward wave oscillates at a single complex frequency, $a_f(z,t)=\tilde{a}_f(z)\exp(i\Delta\omega t)$, where $\Delta\omega$ is to be determined self-consistently from the boundary conditions at the mirror surfaces.

The ponderomotive potential in terms of the time asymptotic field coefficients is given by,

$$\vec{A}_w\cdot\vec{A}=\frac{B_w}{2k_w}\tilde{a}_f(z)\exp\{-i[Kz-(\omega_0+\Delta\omega)t]\}+\text{nonresonant terms}. \tag{A12}$$

Retaining only the phase resonant terms, the formal solution to the linearized Vlasov equation can be written as follows,

$$\tilde{g}^{(1)}(z,p_z,t)=\int_0^z dz'\frac{em\gamma_0}{p_z}\exp\left\{-i\left[K-\frac{(\omega_0+\Delta\omega)}{v_z}\right]z'\right\}\frac{\partial\tilde{g}^{(0)}(p_z)}{\partial p_z}$$

$$(\partial/\partial z'-iK)\left[\frac{\beta_w\overline{\gamma}}{2\gamma_0}\tilde{a}_f(z')-\tilde{\phi}(z')\right]. \tag{A13}$$

Inserting this result into the Vlasov-Maxwell system of equations one obtains,

$$\left\{\frac{\partial}{\partial z}+i\left[\frac{\Delta\omega}{c}-\frac{\omega_p^2}{2\omega_0 c\overline{\gamma}}(1-\beta_w^2/2)\right]\right\}\tilde{a}_f(z)=$$
$$-i\frac{\omega_p^2}{2\omega_0 c\overline{\gamma}}\frac{\beta_w(1-\beta_{z0})}{\beta_{z0}^3}\frac{(\omega_0+\Delta\omega)}{c}\int_0^z dz'(z'-z)\exp\{-i\Delta K(z'-z)\}$$
$$(\partial/\partial z'-iK)\left[\frac{\beta_w}{2}\tilde{a}_f(z')-\tilde{\phi}(z')\right]+\frac{i\omega_p^2}{2\omega_0 c\overline{\gamma}}\frac{\beta_w}{\beta_{z0}^2}$$
$$\int_0^z dz'\exp\{-i\Delta K(z'-z)\}(\partial/\partial z'-iK)\left[(1+\beta_{z0}^2)\beta_w\tilde{a}_f(z')/2-\tilde{\phi}(z')\right], \qquad (A14)$$

$$\left(\frac{\partial}{\partial z}-\frac{i}{2}K\right)\tilde{\phi}(z)=-\frac{\omega_p^2}{2Kc^2\overline{\gamma}}\frac{(1-\beta_{z0}^2)}{\beta_{z0}^3}\frac{(\omega_0+\Delta\omega)}{c}$$
$$\int_0^z dz'(z'-z)\exp\{-i\Delta K(z'-z)\}(\partial/\partial z'-iK)\left[\beta_w\tilde{a}_f(z')/2-\tilde{\phi}(z')\right]$$
$$+\frac{i\omega_p^2}{2Kc^2\overline{\gamma}}\frac{1}{\beta_{z0}^2}\int_0^z dz'\exp\{-i\Delta K(z'-z)\}(\partial/\partial z'-iK)\left[\beta_w\tilde{a}_f(z')/2-(1-\beta_{z0}^2)\tilde{\phi}(z')\right], \qquad (A15)$$

where we have defined $\Delta K=K-(\omega_0+\Delta\omega)/v_{z0}$. The previous set of equations yields a dispersion relation which we shall refer to as the complete dispersion relation. A simplified dispersion relation is obtained by noting that in the momentum integration , the results are most sensitive to changes in the exponent, $\Delta K(z'-z)$. Retaining only the terms in the integration by parts which are proportional to $\partial\Delta K/\partial p_z$, one obtains the following simplified system of equations,

$$\left\{\frac{\partial}{\partial z}+i\left[\frac{\Delta\omega}{c}-\frac{\omega_p^2}{2\omega_0 c\overline{\gamma}}(1-\beta_w^2/2)\right]\right\}\tilde{a}_f(z)$$
$$=\frac{-\omega_p^2}{2\omega_0 c\overline{\gamma}}\frac{\beta_w(1-\beta_{z0}^2)}{\beta_{z0}^3}\frac{(\omega_0+\Delta\omega)}{c}\int_0^z dz'(z'-z)\exp\{-i\Delta K(z'-z)\}$$
$$\left(\frac{\partial}{\partial z'}-iK\right)\left[\frac{\beta_w}{2}\tilde{a}_f(z')-\tilde{\phi}(z')\right], \qquad (A16)$$

$$\left\{\frac{\partial}{\partial z}-\frac{i}{2}K\right\}\tilde{\phi}(z)=-\frac{\omega_p^2}{2Kc^2\overline{\gamma}}\frac{(1-\beta_{z0}^2)}{\beta_{z0}^3}\frac{(\omega_0+\Delta\omega)}{c}$$

$$\int_0^z dz'(z'-z)\exp\{-i\Delta K(z'-z)\}\left(\frac{\partial}{\partial z'}-iK\right)\left[\frac{\beta_w}{2}\tilde{a}_f(z')-\tilde{\phi}(z')\right]. \qquad (A17)$$

In both the cases of the complete and simplified set of equations, the equations are of the convolution type and can be solved by Laplace transform methods. The text of the paper consists of a detailed analysis of the simplified set of equations in the Compton regime.

MED

8